[MSG+22] M.- A. Meyer, S. Silberg, C. Granrath, C. Kugler, L. Wachtmeister, B. Rumpe, S. Christiaens, J. L. Andert:
Scenario- and Model-Based Systems Engineering Procedure for the SOTIF-Compliant Design of Automated Driving Functions.
In: 2022 IEEE Intelligent Vehicles Symposium (IV'22), pp. 1599-1604, IEEE, Juni 2022.
www.se-rwth.de/publications/

Scenario- and Model-Based Systems Engineering Procedure for the SOTIF-Compliant Design of Automated Driving Functions

Max-Arno Meyer, Sebastian Silberg, Christian Granrath, Christopher Kugler, Louis Wachtmeister,
Bernhard Rumpe, Sébastien Christiaens and Jakob Andert*

***Abstract*— Advances in automated driving are creating new challenges for product development in the automotive industry and continuously driving up the cost of product verification and validation. Modern automated driving systems (ADS) must safely handle a considerable number of driving scenarios in compliance with the Safety of the Intended Functionality (SOTIF) standard. While model-based systems engineering (MBSE) has successfully proven itself in the automotive industry as an enabler for complex system and test design, common procedures are neither scenario-based nor do they consider SOTIF. It is yet to be shown, how MBSE approaches can meet these specific requirements of ADS development and, what advantages they can offer over non-model-based methods.**

In this paper, an extended variant of the established feature-driven MBSE procedure CUBE is presented that includes the analysis of use cases and scenarios. Use-case-specific logical scenarios and the corresponding expected behavior and system architecture are specified using SysML profile extensions. It is demonstrated, how specification model artifacts are used for identifying potentially hazardous scenarios and functional deficiencies and how SOTIF analysis results flow back into the specification process by means of the function "Multi-Story Car Park Chauffeur". The SysML model is linked to a safety argumentation created using the Goal Structuring Notation to integrate the system specification and the evidence from the SOTIF analysis in a single procedure and toolchain, ensuring full traceability.

I. INTRODUCTION

Current vehicle development is strongly characterized by automated driving (AD). Alongside the transition to electric powertrains, vehicle automation is a key to safer, more efficient and more environmentally friendly road traffic. It has reached a difficult milestone with the elimination of the need for constant driver supervision in systems classified as SAE Level 3 or higher [1], [2]. The shift in responsibility from human to the machine has major implications for the development strategy and effort, even if a function's driving tasks remain unchanged. Without the driver as a fallback level, a system must be designed and tested to handle driving tasks within its defined operational design domain (ODD), including driver handovers, safer than human drivers [3]. To achieve this, it is indispensable to formally analyze use cases and driving scenarios for the specification of the expected system behavior, the design of the operating principle and the assessment of technical capabilities [4]. In the following this is referred to as scenario-based systems engineering (SE). The development standards to ensure safely operating automated driving systems (ADS) are constantly evolving. This concerns in particular the Safety of the Intended Functionality (SOTIF) standard, which aims to minimize the number of unknown and unsafe scenarios and has been updated recently to version ISO/DIS 21448:2021 [5]. In light of the amended requirements of the standard, new, compliant methods for system design and test must be developed [6].

Model-based systems engineering (MBSE) is becoming increasingly important for the development of automotive applications [7]. By formalizing requirements and architecture specifications using standardized graphical languages and storing it in a single model, MBSE enables complexity control, artifact traceability, reuse, automated test case generation and communication for interdisciplinary development [7], [8]. A comprehensive extension of established MBSE approaches to scenario-based development for automated vehicles is currently still a subject of research. It could provide a significant contribution to efficient development processes for automated vehicles while maintaining conformity to standards. Up to this point, it has not been demonstrated how MBSE can be carried out scenario-based to establish traceability between use cases, scenarios, and expected behavior with respect to SOTIF standard requirements, as well as enable test scenario generation.

This paper provides a meaningful and novel contribution to current research by proposing one of the first technical realizations for the integration of model-based and scenario-based SE methods and for conducting the SOTIF scope at the time of system design by leveraging the advantages of MBSE. In this respect, it addresses the following research questions.

- RQ1: How can a feature-driven MBSE approach be extended for scenario-based SE combining use case, scenario, operating principle and architecture specification in a single SysML model?
- RQ2: How to use specification model artifacts to identify potentially hazardous scenarios and functional deficiencies and link them to SOTIF evidence?
- RQ3: How to feed results of the SOTIF analysis back into the specification model?

The research questions will be addressed in several steps starting with a state-of-the-art analysis of scenario-based specification methods and an overview of the baseline MBSE

---

*M.-A. Meyer and J. Andert are with the Teaching and Research Area Mechatronics in Mobile Propulsion, RWTH Aachen University, 52074 Aachen, Germany. (e-mail: meyer_max@mmp.rwth-aachen.de). S. Silberg, C. Granrath, C. Kugler and S. Christiaens are with the FEV.io GmbH, 52078 Aachen, Germany. L. Wachtmeister and B. Rumpe are with the Chair of Software Engineering, RWTH Aachen University, 52074 Aachen, Germany.

methodology used. A set of requirements for the MBSE approach and the link between specification model and SOTIF analysis will be presented, followed by the description of the methodology extensions. The exemplary demonstration of the procedure shown in this paper refers to an SAE level 3 autonomous parking feature and provides insight into relevant driving scenarios and development decisions.

II. RELATED WORK

Multiple approaches for scenario-based SE have already been proposed [9], [10], [11]. One of the most prominent methods was developed in the PEGASUS funded project [12]. A procedure to achieve the safety goals for ADS including the determination of requirements and scenarios, the processing of scenarios in a database and a test strategy based on scenarios is presented [12], [13]. For machine readable scenario specification, domain-specific languages (DSLs), such as ASAM OpenSCENARIO [14], are commonly used in scenario-based SE methods [15].

The latest revision of the SOTIF standard ISO/DIS 21448:2021 [5] could not be considered by approaches for scenario-based SE published previously. Moreover, the use of MBSE to formalize use case and scenario analysis, system requirements and architecture in a single model, using semi-formal languages such as SysML, was not in focus. Various MBSE approaches have been proposed for application in the automotive industry such as Specification Method for Architecture, Design and Test (SMArDT) [16] or Software Platform Embedded Systems (SPES XT) [17]. These are tailored to system-based development, whereas ADS development is characterized by end-to-end functions at vehicle level (feature), that are distributed across various subsystems and components [8]. The Compositional Unified System-Based Engineering (CUBE) methodology supports MBSE and combines proven methods from the field of SE, such as stepwise partitioning of the system and structured specification by means of different system views, with the agile procedure of feature-driven development (FDD) [8]. By combining these methods, it is possible to reduce system complexity in the specification to a greater extent than is possible with non-feature-based development [8]. The consideration of FDD for the specification of each individual decomposition element enables a functionally focused view of the system [8]. CUBE aims for a solution-neutral consideration of the system that contributes to a cost-optimized development and production of new systems, since organizational and development history related decisions are not considered during the system design [8]. It can be used independently of the system and has already been successfully validated in previous studies, for example in the development of simulation models and vertical take-off and landing aircrafts [18], [19]. As such the solution-neutral and function-oriented view of CUBE's feature-driven approach fits the characteristics of ADS development. However, none of these MBSE approaches include a systematic analysis of scenarios.

To address these aspects, the procedure presented in this paper aims to combine scenario-based SE and MBSE by extending an existing methodology. As described in [20], scenarios are the drivers of the presented approach. However, scenarios here are not intended to replace the behavioral specification concepts presented in other MBSE methods, but rather a complement of those methods to additionally meet the requirements of SOTIF. Therefore, another focus of this paper lies on the benefits of using such a scenario-based MBSE procedure to address the revised SOTIF standard.

III. MODEL-BASED SYSTEMS ENGINEERING FOR SOTIF-COMPLIANT DESIGN OF AUTOMATED DRIVING FUNCTIONS

A. Procedure Requirements

In a first step, a set of requirements is defined for the extensions to state-of-the-art MBSE approaches that are necessary for scenario-based SE of automated vehicles. The development of automated vehicles cannot be achieved on a component basis since the goal is to develop features at vehicle level. To realize an automated driving function, different subsystems such as sensors, electronic control units (ECUs) for decision making and planning as well as actuators for vehicle control are necessary, which must fulfill requirements and interface contracts of many other features in addition to its own. The selected MBSE procedure should therefore be feature-driven to represent AD functions as features of interest at the vehicle level and to be able to traceably break down requirements to subsystems and components.

To enable scenario-based system design using MBSE, a feature's ODD and driving scenarios shall be included in the specification model extending the use case definition which is commonly used in MBSE approaches. An extension profile for the applied modeling language, e.g., UML or SysML, with new diagrams and model elements is needed to model ODDs and scenarios. Modeled scenarios shall represent a machine-readable definition of the time-dependent interaction of the controlled vehicle and its environment. Both, logical scenarios describing parameter spaces in the state space and concrete scenarios describing one representative of this state space are to be able to be modelled in this way [12]. The 6-layer model for a structured description of traffic and environment [21] serves as a baseline for developing the modelling extension. Within the specification model connecting scenarios to use cases shall be enabled, following the SOTIF definition that use cases include scenarios [5]. In addition, a way shall be provided to link the expected behavior of the AD feature to the scenarios with the corresponding triggers for that behavior. The formal representation of use cases, scenarios, expected behavior, architecture, and the traceability among these artifacts in a single model are intended to act as a basis for different SOTIF analysis steps, including identifying potentially hazardous scenarios at design time and transforming these to known safe scenarios by analysis of trigger conditions and system elements. For this purpose, a method to link MBSE model elements as evidence to a SOTIF argument is required. Conversely, it shall also be possible to extend the scenario and system specification iteratively according to the SOTIF analysis results.

B. Extension of the CUBE Methodology for Scenario-based Systems Engineering and SOTIF Analysis

To address the derived requirements, the MBSE methodology CUBE is extended (see Figure 1). As AD features depend on both hardware and software, SysML is used as the modeling language for this new MBSE procedure. All diagram and model element extensions are developed as

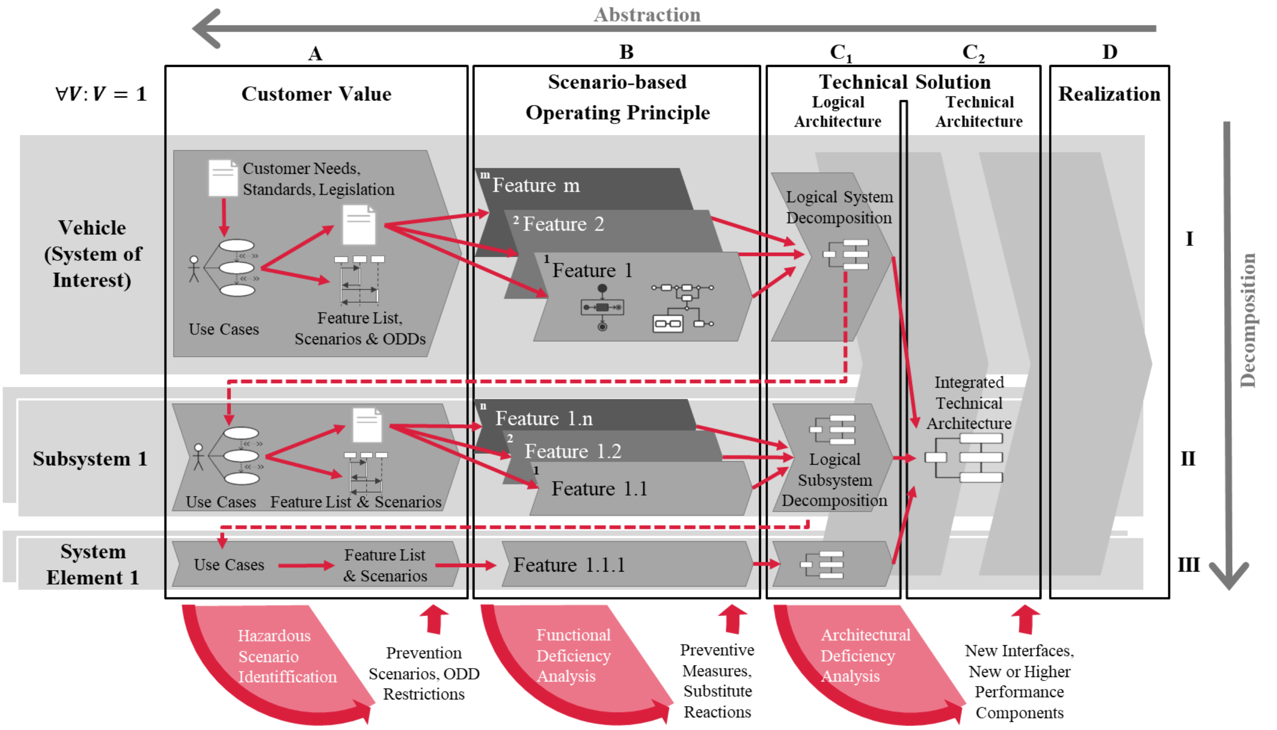

Figure 1. Extended CUBE procedure for SOTIF compliant design of automated driving functions

SysML extensions using stereotyping. The controlled vehicle is defined as the system of interest (SoI) because customer and legislative requirements for AD features relate to the vehicle level and the realization is distributed across multiple subsystems in the vehicle. The basic Z-shape of CUBE's procedure model, where on each decomposition level three abstraction levels are passed through before a further decomposition takes place, is adopted. The customer value contains black-box requirements including use case, scenario, and feature definition, where features represent AD functions such as Car Park Chauffeur or Highway Pilot. The scenario-based operating principle contains the specification of the expected system behavior with respect to each feature. The logical architecture results from the assignment of activities and actions to logical subsystems, e.g., leading to the commonly applied sense-plan-act building blocks for ADS development [4]. The defined logical elements represent the subsystems on the next decomposition level which will again be specified on the same abstraction levels including a new feature definition. Further decomposition can be carried out until logical subsystems can be mapped to technical components such as sensors, ECUs or software components to generate a technical architecture.

To perform the necessary use case and scenario analysis, abstraction layer A, which provides a black-box view on the system, is extended. In this context, black-box view is meant to be a view of the system, not considering the internal structure and operation. SysML use case diagrams are utilized to define vehicle level use cases and their interdependencies using '*include*' and '*extend*' relationships. Use cases are also associated with actors outside the controlled vehicle which the vehicle interacts with for the given use case. Based on the CUBE methodology, use cases are allocated to features. Following the SOTIF definition use cases are enhanced with scenario specifications. To achieve this, a new stereotype '*scenario*' is introduced, which can be connected to use cases or features using an '*include*' relationship and to other scenarios using a '*generalize*' relationship. For each use case and feature, one included scenario is modeled that represents the state space of its specific ODD. Additionally, special scenarios can be defined and linked to the superordinate scenario using '*generalize*' connectors. Modeled scenarios are refined with a textual and graphical description (functional scenario) as well as a set of diagrams specifying the temporal behavior and parameter space (logical scenario) [22]. For the latter a SysML extension profile was developed. A sequence diagram (see Figure 1, abstraction layer A) is used to divide a scenario into phases which define a set of constraints for static environment, objects and actors, environmental conditions, and information technology for a period, and optionally a sequence of maneuvers which define the motion of the occurring objects and the expected behavior of the controlled vehicle over time. Each phase and maneuver are parametrized using block definition diagrams. New block stereotypes are defined and linked using composition connectors to structure information according to the 6-layer model according to Pegasus [21], e.g., a phase features a road, that features road sections, that feature lanes and a topology. Additional dependency connector stereotypes are used to relate road elements or reference objects. Each block stereotype contains several tagged values to parametrize the corresponding information by specification of parameter ranges or concrete values. The new scenario modeling language based on SysML can be used to export standardized scenario specifications in ASAM OpenSCENARIO format and to directly relate the content of a scenario to other MBSE model elements.

As shown in Figure 1, on abstraction layer B, model-based white-box requirements are defined for each feature and use case to specify its operating principle based on the modeled scenarios. To achieve this, logical scenarios or parts of it such as phases, maneuvers or individual parameters are referenced in the guard conditions of state charts and activity diagrams that represent the expected behavior. In this way, a direct cause-effect traceability between logical scenario and expected behavior is established which continues to the architecture following the CUBE procedure.

This traceability can be effectively used for the SOTIF analysis resulting in update iterations of the specification model. Based on the modeled logical scenarios potentially hazardous scenarios can be identified in the state spaces, not only on vehicle feature level, but for each use case from vehicle to component level. This can either be achieved through knowledge driven analysis (known hazardous scenarios) or through test data driven analysis (previously unknown hazardous scenarios). In both cases a risk assessment is performed considering the probability of occurrence of trigger conditions and the safety impact. Subsequently for each identified hazardous scenario a prevention scenario is derived containing the expected safe kinematic behavior of the controlled vehicle for the same trigger conditions and is modeled as a specialized SOTIF scenario for the corresponding use case. Alternatively, a hazardous scenario can be excluded from the ODD and logical scenarios for SAE level 2 systems or level 3 systems, in case a timely safe handover to the driver is possible. Using the traceability within the specification model, use cases and features impacted by the added prevention scenarios are identified on all decomposition levels. The operating principles, system elements and interfaces which can be traced back to these scenarios via model references are analyzed with respect to functional and architectural deficiencies and are iteratively modified to match the expected behavior on vehicle level for the SOTIF scenarios and to transform the potentially hazardous scenarios into known safe scenarios. When the safety case is also conducted model-based using Goal Structuring Notation (GSN) as proposed in the informative annex of ISO21448, added or updated SysML model elements can directly be linked as evidence.

IV. DEMONSTRATION OF THE PROCEDURE BASED ON THE FUNCTION MULTI-STORY CAR PARK CHAUFFEUR

A. *Initial System Specification*

The extended procedure is demonstrated for the vehicle (SoI) level by means of the feature '*Multi-Story Car Park Chauffeur'* and the allocated exemplary use case '*parking autonomously'* (see Figure 2). This use case represents an extension to the superordinate use case *driving autonomously in a car park*. The ODD specification that defines the framework conditions under which the feature to be developed is to be designed and validated is applicable to all use cases. Accordingly, with respect to the application example, the ODD model defines the specifics of a multi-story car park environment including types of parking lots, connection roads, traffic signs and rules such as the right-of-way rule, and multiple dynamic objects like cars, pedestrians and shopping carts. Restricting the ODD to the particular use case provides the use case specific logical scenario *'SC_L1_Parking autonomously'*. For '*parking autonomously'* an empty parking lot, the access road section and optional adjacent parking lots are parametrized with respect to topology and lane markings in a phase block. Additionally, the entities for this scenario including the controlled vehicle which is initialized on the access road as well as potential other parking or driving vehicles and pedestrians are specified. Using a maneuver block entering the parking lot is modeled as the expected behavior related to this use case while the motion of other moving objects is arbitrary for this general case.

For the feature '*Multi-Story Car Park Chauffeur', which* the use case is allocated to, the operating principle is modeled. In this case an activity diagram is used that includes a variety of activities linked to the different use cases, which can be allocated to the logical architecture elements '*sense'*, '*plan'*, and '*act'* that are further specified on the next decomposition level. Other behavior diagrams, such as sequence diagrams, can be used to analyze and specify timing behavior, but a detailed consideration is omitted for the purposes of this paper. Through iterative decomposition considering the different levels of abstraction, the relevant system elements, technical components, and a more detailed operating principle are identified step by step and therefore the best possible knowledge of the intended functionality is achieved. The relevant activities for this demonstrator are the ones contributing to the expected behavior when parking autonomously such as the detection of lanes, empty parking lots, signage, obstacles, driving and parking vehicles and pedestrians, the decision making to start parking as well as the trajectory planning and vehicle control for parking.

B. *SOTIF Analysis based on the System Model and Corresponding Model Modifications*

The identification and evaluation of hazards is the first step of the SOTIF analysis performed after the system specification. In relation to the logical scenario of '*parking autonomously'*, for example, the opening of a car door of an adjacent car can be identified as a hazardous event. Without a preventive system behavior this event leads to a potentially hazardous scenario. The a-priori worst case conditions of such an event must be determined as well as their likelihoods [5]. Considering the likelihood of this event trigger in a car park and the potential consequence of a collision, the resulting risk of harm is not acceptable in this case. If it was identified as acceptable, this identified hazard would not be considered any more. Since the severity of the hazardous event is unacceptable, controllability must be evaluated. The identified hazardous event is modeled in a preventive scenario *'SC_L2_SOTIF_Reacting to the opening of a door'* including the desired kinematic behavior, representing a special case of the use cases' logical scenario, which is connected via a "generalize" relationship as described in section III-B (see Figure 2). This step transforms the known hazardous into a known non-hazardous scenario. The operating principle and the system architecture need to be modified in a way, that the desired safe behavior of the system, described in the preventive scenario, is achieved. In this case, the awareness of the system to protruding objects such as opened doors while parking is specified in the operating principle of the feature as shown in Figure 2. The only safe response to such an event is stopping the vehicle to interrupt the parking process and therefore also implemented.

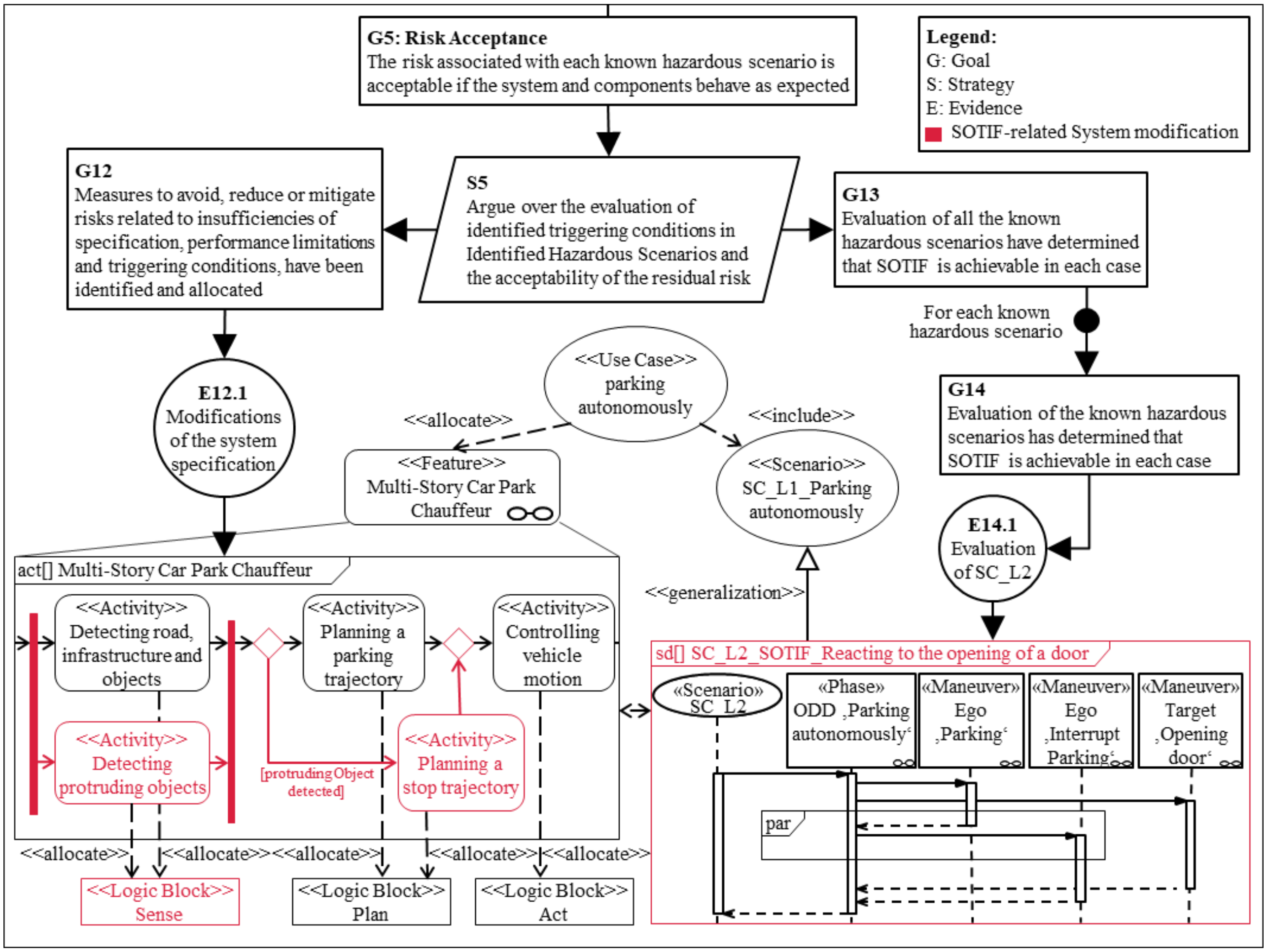

Figure 2. Exemplary model artifacts of the extended CUBE procedure (layer A to C1) linked as evidence to SOTIF GSN elements as defined in Ref. [5]

If a preventive scenario and behavior for the identified hazardous event are already specified, an unacceptable risk of harm can still not be excluded, as the trigger conditions need to be analyzed to identify functional insufficiencies. This analysis is performed by the identification of the systems elements potentially leading to SOTIF-related hazardous events [5]. The corresponding feature and its activities lead to the allocated architectural elements and their interfaces. By allocating relevant known potential functional insufficiencies to the system elements and interfaces, the overall functional insufficiency can be identified in each hazardous case. For example, the allocated logical component '*sense*' (see Figure 2), whose further specification ultimately leads to underlying technical components such as sensors, is responsible for the activity to detect an adjacent car's door opening. Further analysis may reveal that the used sensor setup impairs the expected behavior of the activity with respect to detection area or the ODD specific lighting conditions. The triggering condition effect is a delayed perception of the dangerous situation and therefore there is still a risk of collision. A comparable analysis can be conducted for every activity and the linked architectural elements enabled by the traceability inside the SysML model. Based on the analysis of system elements another iterative update of the model-based specification is performed, introducing measures to improve the SOTIF. These measures represent changes to the system architecture and subsequent decomposition layers. In case of the known functional inadequacy of insufficient perception of intruding objects during parking by the sensor set, it must be adjusted, e.g., by selecting higher performance sensors to ensure continuous perception under the operating conditions defined in the corresponding scenario. This requirement is the starting point for the subsequent iteration of the specification.

C. Model-based SOTIF-Evidence and Safety Argument

To consider the SOTIF as fulfilled, it is necessary to link the elements of evidence into a chain of evidence. This can be done by means of the GSN, with the help of which a complete safety argumentation has already been specified in the non-normative annex of SOTIF [5]. Via an allocation of the elements of evidence to the derived safety objectives, the SOTIF can be reasonably justified as having been met. Various modeling tools, e.g., Enterprise Architect, which is used for this demonstrator, allow to link artifacts of a SysML model and a GSN. This provides an option to formally integrate the documentation of the SOTIF argument and the presented, extended MBSE procedure. Additionally, this method is supported by the risk analysis and Assessment Modeling Language as an extension to the SysML [21]. Figure 2 shows an example of how specification artifacts from abstraction layers A to C1 can be used as evidence elements. The specification of the prevention scenario for the known dangerous scenario when parking autonomously on layer A is the basis for the consideration of the scenario in the subsequent SOTIF analysis and is therefore one of the evidence elements for goal 14. The analysis results are fed back into the specification as modifications on abstraction layers B and C. Goal 12 aims to keep track on these taken measures in the system specification to achieve the SOTIF and is therefore linked to the corresponding realized system modifications in the operating principle and architecture. This ensures that known hazardous scenario have been analyzed and changes to the system specification to transform it into a known safe scenario have been implemented accordingly.

V. CONCLUSION AND OUTLOOK

The main objective of this work is to develop a MBSE procedure that supports scenario-based SE to meet the requirements of ADS development. An extension for the feature-driven SE methodology CUBE has been developed for this purpose (RQ1). AD function requirements can be modeled across different abstraction and decomposition layers. The new procedure enables a formal use case and scenario analysis as well as modeling logical scenarios using SysML. The trigger conditions in modeled scenarios can be directly linked to the expected system behavior that is specified in behavior diagrams. Thus, full traceability between scenarios and system requirements is achieved by storing all information in a single model. It has been successfully demonstrated by the example of the use case *'parking autonomously'*, how this traceability can be used for a SOTIF assessment (RQ2). Potentially hazardous scenarios can be identified based on use case specific ODDs and prevention scenarios can be modeled. Functional deficiencies in the operating principle and system elements relevant to specific scenarios can be tracked and corrected (RQ3). A direct allocation of SysML artifacts to a SOTIF GSN as evidence has been presented. Thus, the core contribution is a new procedure that integrates MBSE, scenario-based SE and SOTIF analysis for the first time.

The next step is to validate the procedure considering the final release of ISO21448:2021. Finally, some aspects worthy of further investigation were identified, such as exchanging scenarios between the SysML model and scenario databases and generating logical test cases directly from the model.

REFERENCES

[1] T. Hirose, T. Sawaragi, H. Nomoto, and Y. Michiura, “Functional safety analysis of SAE conditional driving automation in time-critical situations and proposals for its feasibility,” Cogn Tech Work, vol. 23, pp. 639–657, 2021.

[2] SAE International, “J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” 2018

[3] M. A. Nees, “Safer than the average human driver (who is less safe than me)? Examining a popular safety benchmark for self-driving cars,” Journal of Safety Research, vol. 69, pp. 61-68, 2019.

[4] M. Wood, D. Wittmann, S. Liu, C. Knobel, S. Syguda, T. Wiltschko, N. Garbacik, M. O‘Brien, U. Dannebaum, J. Weast, B. Dornieden, et al., „Safety first for automated driving,” 2019, https://www.daimler.com/documents/innovation/other/safety-first-for-automated-driving.pdf, accessed 10. January 2022.

[5] International Organization for Standardization, “ISO/DIS 21448:2021, Road vehicles - Safety of the intended functionality,” 2021.

[6] M. Meyer, L. Sauter, C. Granrath, et al., ”Simulator Coupled with Distributed Co-Simulation Protocol for Automated Driving Tests,” Automot. Innov. vol. 4, pp. 373–389, 2021.

[7] S. Kriebel, J. Richenhagen, C. Granrath and C. Kugler, “Systems Engineering with SysML - The Path to the Future?,” MTZ worldwide, vol. 79, pp. 44–47, 2018.

[8] C. Granrath, C. Kugler, S. Silberg, M.-A. Meyer, et al. „Feature-driven systems engineering procedure for standardized product-line development,” Systems Engineering, vol. 24, pp. 456-479, 2021.

[9] C. Sippl, F. Bock, C. Lauer, A. Heinz, T. Neumayer and R. German, "Scenario-Based Systems Engineering: An Approach Towards Automated Driving Function Development," 2019 IEEE International Systems Conference (SysCon), pp. 1-8, 2019.

[10] I. Graessler, J. Hentze and P. Scholle, "Enhancing systems engineering by scenario-based anticipation of future developments," 2016 11th System of Systems Engineering Conference (SoSE),, pp. 1-5, 2016.

[11] M. Kremer, S. Christiaens, C. Granrath, et al., “Scenario- and Model-based Systems Engineering for Highly Automated Driving,” ATZ Worldw, vol. 122, pp. 16–21, 2020.

[12] Deutsches Zentrum für Luft- und Raumfahrt e. V., “Schlussbericht für das Gesamtprojekt PEGASUS. Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen,“ 2020.

[13] A. Pütz, A. Zlocki, J. Bock and L. Eckstein. “System validation of highly automated vehicles with a database of relevant traffic scenarios.”, 2017.

[14] Association for Standardization of Automation and Measuring Systems, “ASAM OpenSCENARIO: User Guide,” 2021, https://www.asam.net/index.php?eID=dumpFile&t=f&f=4596&token=55bca7d8439f2bae072c4dffd1ee544a6d76b786, accessed 23 January 2022.

[15] F. Bock, C. Sippl, A. Heinz, C. Lauer and R. German, "Advantageous Usage of Textual Domain-Specific Languages for Scenario-Driven Development of Automated Driving Functions," 2019 IEEE International Systems Conference (SysCon), pp. 1-8, 2019.

[16] S. Kriebel, V. Moyses, G. Strobl, J. Richenhagen, P. Orth, S. Pischinger, et al., “The next generation of bmw’s electrified powertrains: Providing software features quickly by model-based system de-sign,” 26th Aachen Colloquium Automobile and Engine Technology, Aachen, 2017.

[17] K. Pohl, M. Broy, H. Daembkes, and H. Hönninger, “Advanced Model-Based Engineering of Embedded Systems: Extensions of the SPES 2020 Methodology,” Springer International Publishing, Cham, ISBN 978-3-319-48003-9, 2016.

[18] C. Granrath, M. Meyer, J. Andert, J. Ewald, et al. “EleMA: a reference simulation model architecture and interface standard for modeling and testing of electric vehicles,” eTransportation. 2020.

[19] N. Jäckel, C. Granrath, L. Wachtmeister, A. Karaduman, et al. „Feature-Driven Specification of VTOL Air-Taxis with the Use of the Model- Based System Engineering (MBSE) Methodology CUBE,” Vertical Flight Society’s 77th Annual Forum & Technology Display, 2021.

[20] M. Dalibor, N. Jansen, B. Rumpe, L. Wachtmeister and A. Wortmann, "Model-Driven Systems Engineering for Virtual Product Design," 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion, pp. 431-436, 2019.

[21] M. Scholtes et al., "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment," in IEEE Access, vol. 9, pp. 59131-59147, 2021.

[22] M. Meyer, C. Granrath, J. Wefers, et al., "Verfahren zum Ermitteln von Szenarien zum Testen, Validieren und/oder Spezifizieren von Fahrzeugen und/oder Fahrzeugkomponenten," German Patent DE102021004327A1, Dec. 2, 2021.

[23] Object Management Group, “Object Management Group RAAML Beta Version 1.0 Specification Defines Extensions to SysML”, Boston, MA, 2021, https://www.omg.org/news/releases/pr2021/09-15-21.htm, accessed 24. January 2022