Faculty of Computer Science, Electrical Engineering and Mathematics

Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering

PhD Thesis
to obtain the degree of
Doktor der Naturwissenschaften (Dr. rer. nat.)

by
M.Sc. Sergej Japs

Referees:
Prof. Dr.-Ing. Roman Dumitrescu
Prof. Dr. rer. nat. Frank Kargl

Paderborn, June 2024

Acknowledgements

This dissertation was written during my work as a Research Associate at the Fraunhofer Research Institute for Mechatronic Systems Design IEM in collaboration with the University of Paderborn. It is the result of my scientific work in the context of research and industrial projects.

Above all, I thank God, the Father, the Son and the Holy Spirit, for answering my prayer in the summer of 2003. I just wanted to have the opportunity to continue going to school and eventually study. You made much more of it.

I would like to express my thanks to Prof. Dr.-Ing. Roman Dumitrescu for the structured training. I would especially like to thank him for the training according to the T-model. I was able to learn to think in large contexts and to conduct demand-oriented research. At the same time, I was able to deepen my knowledge in systems engineering.

I would like to thank Prof. Dr. rer. nat. Frank Kargl for the exciting collaboration and the many valuable discussions during the joint research projects. My special highlight was the joint research trip to Japan.

I would like to thank Prof. Dr. Eric Bodden for the discussions and the comments on my work. Your advice helped me to ensure the continuity of my work.

I would like to thank my department head Dr.-Ing. Harald Anacker for the trust he has placed in my work. I thank you, Harald, that I was able to discuss even difficult topics with you and that we always found a solution.

I would also like to thank all my colleagues for the pleasant working atmosphere. I am especially grateful for the constructive discussions in the context of my dissertation: Prof. Dr. Lydia Kaiser, Dr. Christian Koldewey, Dr. Matthias Meyer, Dr. Jörg Holtmann, Aschot Kharatyan, Rik Rasor and Daria Wilke. Furthermore, I would like to thank all the students who have supported me in my work through their theses or through their student assistance.

I would like to especially emphasise the thanks to my family: First and foremost, thanks to my parents and parents-in-law.

However, my biggest thank you goes to my wife Kerstin! Thank you for being there for me all these years. Thank you for our two daughters Victoria and Elisabet. Thank you for your patience, faithfulness and love.

Paderborn, June 2024 Sergej Japs

Abstract

The World Forum for Harmonization of Vehicle Regulations (UNECE WP.29) has issued UN Regulation No. 155. It defines uniform conditions for the approval of vehicles with regard to cybersecurity and the CyberSecurity Management System (CSMS). A CSMS consists of processes to identify, assess and treat cybersecurity risks as part of the vehicle development process. Without a valid CSMS, newly produced vehicles may not be approved in the EU from July 2024.

ISO/SAE 21434 describes detailed requirements for a CSMS. It requires the creation of a cybersecurity concept, consisting of 15 work products to be created. The creation of the cybersecurity concept requires the collaboration of experts from different disciplines. Model-Based Systems Engineering (MBSE) supports the collaboration in the concept phase and helps to build a common system understanding between the subject matter experts.

In this work, I present a framework that supports the creation of the 15 work products of the concept phase using MBSE. The creation of the work products is done by a process model and by using several supporting tools that I have developed. The result is a cybersecurity concept according to ISO/SAE 21434.

The framework was evaluated in three workshops with subject matter experts from the field of automotive security engineering, using the Intelligent Speed Assistant application example. The credibility of the work by the subject matter experts was strengthened by a real-life test with a test vehicle.

Preliminary work

[JRK19] Japs, S., Rasor, R., Kaiser, L., Dumitrescu, R. (2019), Model checking of integratively designed product and production systems, Tag des Systems Engineering, Gesellschaft für Systems Engineering, München, ISBN-13: 9783981880557.

[JKK20] Japs, S., Kaiser, L., Kharatyan, A. (2020), Method for 3D-environment driven domain knowledge elicitation and system model generation. In: Proceedings of the Design Society: DESIGN Conference, vol. 1, 197-206. https://doi.org/10.1017/dsd.2020.41.

[Jap20] Japs, S. (2020), Security & safety by model-based requirements engineering. In: IEEE 28th International Requirements Engineering Conference (RE), 422-427. https://doi.org/10.1109/RE48521.2020.00062.

[JAD21] Japs, S., Anacker, H., Dumitrescu, R. (2021), SAVE: Security & safety by model-based systems engineering on the example of automotive industry. In: Procedia CIRP, vol. 100, 187-192. https://doi.org/10.1016/j.procir.2021.05.053.

[JA21] Japs, S., Anacker, H. (2021), Resolution of safety relevant security threats in the system architecture design phase on the example of automotive industry. In: Proceedings of the design society, vol. 1. https://doi.org/10.1017/pds.2021.517.

[Jap21] Japs, S.(2021), Towards the development of the cybersecurity concept according to ISO/SAE 21434 using model-based systems engineering, In: IEEE 29th International Requirements Engineering Conference (RE), 486-491. https://doi.org/10.1109/RE51729.2021.00073.

[JAK21] Japs, S, Anacker, H., Kaiser, L., Holtmann, J., Dumitrescu, R., Kargl, F.(2021), D-REQs: Determination of security & safety requirements in workshops based on the use of model-based systems engineering. In: IEEE 29th International Requirements Engineering Conference Workshops (REW), 412-414. https://doi.org/10.1109/REW53955.2021.00073.

[JKA22] Japs, S., Kargl, F., Anacker, H., Dumitrescu, R.(2022), Why make it hard? - Usage of aggregated statistical data for risk assessment of damage scenarios in the context of ISO/SAE 21434. In: Procedia CIRP, vol. 109, 293-298. https://doi.org/10.1016/j.procir.2022.05.252.

[JSK22] Japs, S., Schmidt, S., Kargl, F., Kaiser, L., Kharatyan, A., Dumitrescu, R.(2022), Collaborative modeling of use case & and damage scenarios in online workshops using a 3D environment. In: Proceed-

ings of the Design Society: DESIGN Conference, vol. 2, 1599-1608. https://doi.org/10.1017/pds.2022.162.

[JFA23] Japs, S., Faheem, F., Anacker, H., Husung, S., Dumitrescu, R.(2023), Model-based systems engineering using security design patterns in the context of ISO/SAE 21434. Proceedings of the Design Society, 3. https://doi.org/10.1017/pds.2023.268.

[JKA23] Japs, S., Kargl, F., Anacker, H., Bodden, E., Koldewey, C., Dumitrescu, R.(2023), Using model-based sytems engineering to derive requirements from a cybersecurity concept in accordance with ISO/SAE 21434 (Submitted research paper).

[WJK20] Wiecher, C., Japs, S., Kaiser, L., Greenyer, J., Dumitrescu, R., Wolff, C.(2020), Scenarios in the loop: Integrated requirements analysis and automotive system validation, In: Proceedings of the 23rd ACM/IEEE International conference on model driven engineering languages and systems: Companion proceedings, virtual event, Canada, https://doi.org/10.1145/3417990.3421264.

[KKJ21] Kargl, F., Koszescha, J., Japs, S.(2021), Security for connected automated cars, In: 19th escar Europe : The World's Leading Automotive Cyber Security Conference, https://hss-opus.ub.ruhr-uni-bochum.de/opus4/frontdoor/index/index/docId/8345.

[KTJ21] Kharatyan, A., Tekaat, J., Japs, S., Anacker, H., Dumitrescu, R.(2021), Metamodel for safety and security integrated system architecture modeling, Proceedings of the Design Society, 1, 2027-2036, https://doi.org/10.1017/pds.2021.464.

[KGA22] Kharatyan, A., Günther, M., Anacker, H., Japs, S., Dumitrescu, R.(2022), Security- and safety-driven functional architecture development exemplified by automotive systems engineering, Proceedings of the Design Society, 9, 586-591, https://doi.org/10.1016/j.procir.2022.05.299.

[KTH23] Kargl, F., Trkulja, N., Hermann, A., Sommer, F., Andersson, F., Japs, S.(2023), Securing Cooperative Intersection Management through Subjective Trust Networks, In: IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, 2023, pp. 1-7, https://doi.org/10.1017/pds.2021.464.

[JM23] Japs, S., Mundt, E. (2023), Modellbasiertes Anforderungsmanagement, In: Systems Engineering - Das Engineering für die Wertschöpfung von Morgen erfolgreich gestalten, (Book section prepared for publishing).

[JK23] Japs, S. (2023), Security und Safety by Design - Anwendung von Lösungsmustern, In: Systems Engineering - Das Engineering für die Wertschöpfung von Morgen erfolgreich gestalten, (Book section prepared for

publishing).

[J23] Japs, S. (2023), Frühzeitige Absicherung der Sicherheit, In: Systems Engineering - Das Engineering für die Wertschöpfung von Morgen erfolgreich gestalten, (Book section prepared for publishing).

Table of contents

Page

1 Introduction

This dissertation was written during my work as a research associate at the Fraunhofer Research Institute for Mechatronic Systems Design IEM in partnership with the University of Paderborn. It is the result of my scientific work in the context of research and industry projects.

The core of the work is an approach for the development of a cybersecurity concept according to ISO/SAE 21434 Road vehicles - Cybersecurity engineering [**ISO21**]. This approach integrates partial results that I have developed in the context of research and industry projects. Within the scope of several industrial projects with a German premium vehicle manufacturer, I was able to learn about the challenges in the concept phase and to develop solutions for them. In the context of the research projects SecForCARs (Security for Connected Automated Cars) and SAVE (Securing Automated Vehicles), I was able to learn about the challenges in the area of Security-by-Design and develop solutions for them. The SecForCARs project focused on researching methods, procedures and tools for critical in-vehicle communication. The SAVE project was an extension of SecForCARs with a focus on external vehicle communication.

This work is in the topic area of Systems Engineering, specifically Model-Based Systems Engineering (MBSE), and describes a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*.

1.1 Context and motivation

The World Forum for Harmonization of Vehicle Regulations (UNECE WP.29) issued UN Regulation No. 155 in 2021. It defines unified conditions for the approval of vehicles with regard to cybersecurity and the cybersecurity management system (CSMS). A CSMS refers to a systematic, risk-based approach in defining organisational processes, responsibilities, and governance in managing risks related to cyber threats to vehicles and in protecting vehicles from cyber attacks. Approval authorities are only allowed to grant type approvals with regard to cybersecurity for those types of vehicles that comply with UN R155. [**UN21**]

UN R155 started to be mandatory for new vehicle types in the EU in July 2022 and will be mandatory for all newly produced vehicles in the EU from July 2024 [**TU22-ol**]. As a result, vehicle manufacturers and associated suppliers are required to implement a CSMS.

ISO/SAE 21434 (Road vehicles - Cybersecurity engineering) specifies technical requirements for cybersecurity risk management during the development of road vehicles, particularly in the concept phase. The standard defines requirements for cybersecurity processes and a common language for the communication and management of cybersecurity risks [**ISO21**]. ISO/SAE 21434 concretises UN R155 to a great extent providing detailed requirements.

1.2 Problem

UN R155 will be mandatory in the EU for all new vehicle types from July 2022 and for all newly produced vehicles from July 2024. This means that vehicles developed without a valid CSMS cannot be registered in the EU.

ISO/SAE 21434 describes specific requirements for a process framework to ensure cybersecurity in the automotive sector. The implementation of the cybersecurity process framework in the company constitutes the CSMS required by UN R155.

ISO/SAE 21434 describes requirements for activities to identify cybersecurity risks, cybersecurity goals and cybersecurity requirements for a system to be developed as part of the concept phase. In addition, ISO/SAE 21434 describes requirements for Threat Analysis and Risk Assessment (TARA) methods. The result of the concept phase is the cybersecurity concept.

To create the cybersecurity concept, 15 work products must be created that must meet numerous requirements of ISO/SAE 21434. Here, ISO/SAE 21434 describes requirements for creating the work products, but does not define how these work products must be created. These work products must meet numerous requirements and are highly interrelated.

Since the development of complex intelligent and networked systems requires the collaboration of various disciplines [**GRS14**], the creation of the work products taking into account the numerous requirements is very challenging without a concrete procedure.

1.3 Goal

The goal of my work is to develop a framework for the creation of a cybersecurity concept according to ISO/SAE 21434. This is to support the creation of the 15 work products of the concept phase of ISO/SAE 21434 by means of a concrete procedure.

In accordance with ISO/SAE 21434, the approach to be developed shall include creating the item definition (i.e. set of components that implements a function at the vehicle level), conducting the threat analysis and risk assessment (TARA), and deriving the cybersecurity goals and cybersecurity requirements.

According to ISO/SAE 21434, the 15 work products are closely interrelated. In the context of my work, MBSE shall be used for the systematic use of models and requirements in early system design. This is to make complexity manageable, ensure a common understanding of the system between stakeholders in the concept phase, and ensure traceability between work products.

To validate the approach, the 15 work products of the concept phase shall be created using a realistic and continuous example.

1.4 Assumptions

In order to apply my approach in a company, I make the following assumptions:

A1 (Consideration of cybersecurity in product development): In the company, cybersecurity is used in the context of product development, especially the concept phase. This includes the following: (1.1) Responsibilities and authorities exist for performing cybersecurity activities. (1.2) Resources exist for the execution and management of cybersecurity activities. (1.3) Cybersecurity awareness exists, including associated competency management and awareness training.

A2 (Use of Model-Based Systems Engineering): Model-Based Systems Engineering (MBSE) is used in the company as part of product development, especially the concept phase. This includes the following: (2.1) Responsibilities and authorities exist for performing MBSE activities. (2.2) Resources exist for the execution and management of MBSE activities. (2.3) Awareness of model-centric design of complex technical systems exists.

1.5 Overview

In **Chapter 2**, I present the problem analysis. For this purpose, I first introduce UN Regulation 155 (UN R155), which is relevant to this work and requires vehicle manufacturers to establish a Cybersecurity Management System (CSMS) for type approval. A CSMS comprises a set of interrelated processes designed to enhance the cybersecurity of a vehicle. ISO/SAE 21434 Road vehicles - Cybersecurity engineering is then presented. ISO/SAE 21434 defines specific requirements for a CSMS, but does not describe how these requirements should be realized. This is followed by the introduction of the Model-Based Systems Engineering (MBSE) approach and the description of existing domain-specific approaches from the area of security and safety. These approaches serve as the basis for the realization of the requirements for a CSMS in this work. Lastly, the problem delimitation for the framework to be created follows. This includes the identification of challenges in the realization of ISO/SAE 21434 in the concept phase with the help of MBSE. This is followed by the description of the fields of action for my work. Based on the fields of action, I derive the requirements for my work.

In **Chapter 3**, I introduce the research method that underlies my work. My research method is based on an extension of the Design Science Research (DSR) approach, which supports an application-oriented validation. For this purpose, I first describe the DSR approach and relate it to my work. Then, I present an extension of the DSR approach that supports the repeated usage of the DSR approach. This extended approach allows the iterative and incremental improvement of my work.

In **Chapter 4**, an analysis of the state of the art is conducted. For this purpose, existing approaches from the automotive domain are considered, which are used in the context of system design and especially in the concept phase. Thereby, approaches are investigated

that are mainly relevant for the following areas: security, safety and MBSE. Finally, the analyzed approaches are compared with the requirements identified in Section 2. It becomes clear that there is an urgent need for action with regard to the targeted framework.

In **Chapter 5** I describe the core of my work. This includes the description of the *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*. The framework consists of several parts, which are connected by a procedure model. For the realization of a vehicle function to be developed, a risk analysis is necessary. This includes the identification of damage scenarios. In my work, I present a 3D environment with which damage scenarios can be visualized during the concept phase and discussed by an interdisciplinary team of experts. The damage scenarios have to be rated in terms of their criticality. To facilitate the rating, I provide statistical data in aggregated form. To avoid damage scenarios, I present an approach to identify threat scenarios in SysML models. Regarding threat scenarios, it is necessary to identify the affected components and component relationships in a system architecture. In the context of my industrial projects with a German premium car manufacturer, ECML was used as modeling language in the concept phase and SysML in the detailed system design. To keep the modeling effort low, I present an approach for model transformation from ECML models to SysML models. As a prerequisite for resolving threat scenarios, the countermeasures to be applied must be described in terms of requirements. For this purpose, I present an approach that describes the derivation of requirements based on SysML models.

In **Chapter 6** I present the evaluation of my work. In accordance with the research method in Chapter 3, the work is evaluated using four iterations. In each iteration, I describe my experiences, findings, and conclusions that I used to improve my work for the next iteration. In the final iteration, I present the implementation of my approach using a continuous application example. During the first three iterations, the results were evaluated primarily within the context of teaching. The last iteration was evaluated within one year based on three workshops with experts in the field of automotive security.

In **Chapter 7**, I evaluate my work with respect to the stated requirements. In doing so, I summarize the contents and contributions of my work and state the identified limitations. Based on this, I give an outlook on future research areas. The **Appendix** contains supplementary information on the framework.

2 Problem analysis

The goal of the problem analysis is to identify requirements for a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*. For this purpose, I first present the context of my work. The context is on the one hand the UN Regulation 155 and on the other hand the ISO/SAE 21434, which I present in Section 2.1.2. The basis for the realization of my work is the Model-Based Systems Engineering approach (MBSE) and domain-specific approaches from the field of security and safety engineering, which I describe in the Sections 2.2 and 2.3. In Section 2.4, I introduce the problem delimitation and describe the main challenges of my work. Based on this, I derive the fields of action of my work. Finally, based on the problem delimitation, I derive the requirements of my work in the Section 2.5.

2.1 Relevant regulations and standards

2.1.1 UN Regulation No. 155 - Cybersecurity and Cybersecurity Management System

UN Regulation No. 155 (UN R155) [**UN21**] describes uniform conditions for the approval of vehicles with regard to cybersecurity and the cybersecurity management system. Until the publication of UN R155 in 2021, the consideration of cybersecurity in modern cars was a matter of good practice and was not a formal regulatory requirement for applying for vehicle type approval. UN R155 requires the implementation of a Cyber Security Management System (CSMS). A CSMS refers to a systematic, risk-based approach to defining organisational processes, responsibilities and governance for managing risks related to cyber threats to vehicles and protecting vehicles from cyber attacks. In the European Union, this new cybersecurity regulation will be mandatory for all new vehicle types from July 2022 and all newly produced vehicles from July 2024.

As stated in the regulation itself, these new rules are required by the approval authorities to be imposed only on vehicle manufacturers. However, some parts of these regulations concern security aspects throughout the supply chain and therefore also affect each individual supplier of security critical elements (cf. Figure 2-1). The responsibility then lies with the applicants for vehicle type approval to derive the appropriate requirements for their own suppliers. Manufacturers are required to gather a sufficient amount of evidence to demonstrate their capability to develop, operate and maintain the security of the supplied elements throughout the life cycle of the vehicle. Based on an established and approved CSMS, OEMs must provide vehicle type-specific evidence that demonstrates appropriate measures to mitigate cyber risks associated with their products (e.g. consistent risk assessment, relevant mitigation measures, etc.). OEMs must define relevant cybersecurity requirements to pass on to their suppliers to ensure end-to-end security throughout the supply chain.

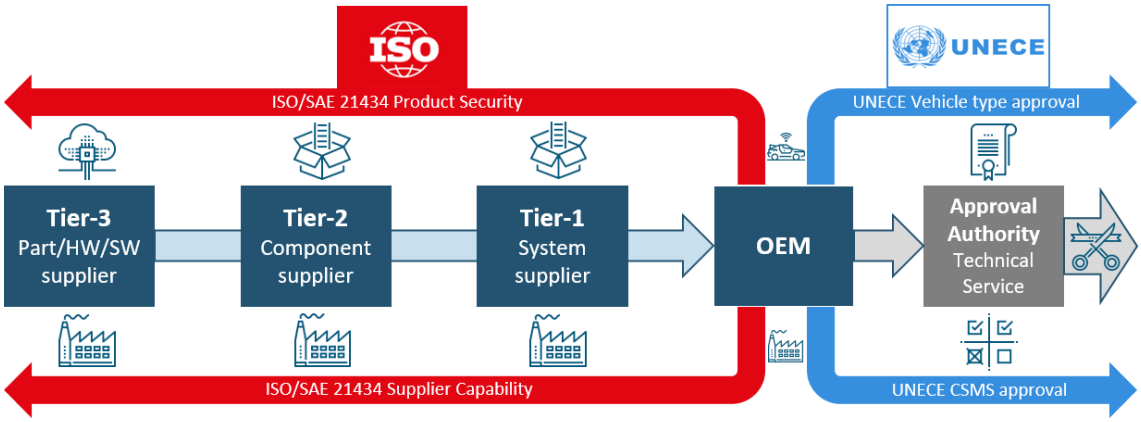


*Figure 2-1: Relationship between UN R155 and ISO/SAE 21434 [***Cer21-ol***].*

According to UN R155, the vehicle manufacturer must prove that the procedures within its cybersecurity management system ensure that the security aspect is adequately addressed. In this context, the threats and mitigation measures from Annex 5 of UN R155 have to be considered.

In order to avoid having to identify threat scenarios over and over again for each project, UN R155 describes a total of 67 threat scenarios for different categories (cf. Excerpt in Figure 2-2) and requires evidence that these threat scenarios were considered in the risk analysis.

Appropriate countermeasures must be selected for critical threat scenarios. UN R155 presents a total of 24 mitigation (cf. Excerpt in Figure 2-2) measures from different categories. Here, as with the threat scenarios, evidence is also required that these have been considered in the risk analysis.

UN R155 specifically requires manufacturers to have skilled personnel with cybersecurity competencies and expertise in automotive risk assessments. Here, reference is made to ISO/SAE 21434. While UN R155 focuses on type approval, ISO/SAE 21434 (cf. Section 2.1.2) describes concrete requirements for a process framework to ensure cybersecurity in the automotive sector. The implementation of the cybersecurity process framework in the company forms the CSMS required by UN R155.

In my work I will present an implementation of the CSMS for the concept phase, which considers the threats and mitigations of UN R155 (cf. Section 5).

2.1.2 ISO/SAE 21434: Road Vehicles - Cybersecurity Engineering

SAE J3061 [**SAE16**] is a guide for vehicle cybersecurity published in 2016 that was developed based on existing industry and government practices and existing conference papers. ISO/SAE 21434 [**ISO21**] was published in 2021 and replaces SAE J3061.

ISO/SAE 21434 specifies technical requirements for cybersecurity risk management related to the concept, product development, production, operation, maintenance and decommis-

Table A1 reference	*Threats to "External connectivity and connections"*	*Ref*	*Mitigation*
16.1	Manipulation of functions designed to remotely operate vehicle systems, such as remote key, immobiliser, and charging pile	M20	Security controls shall be applied to systems that have remote access
16.2	Manipulation of vehicle telematics (e.g. manipulate temperature measurement of sensitive goods, remotely unlock cargo doors)		
16.3	Interference with short range wireless systems or sensors		
17.1	Corrupted applications, or those with poor software security, used as a method to attack vehicle systems	M21	Software shall be security assessed, authenticated and integrity protected. Security controls shall be applied to minimise the risk from third party software that is intended or foreseeable to be hosted on the vehicle
18.1	External interfaces such as USB or other ports used as a point of attack, for example through code injection	M22	Security controls shall be applied to external interfaces
18.2	Media infected with viruses connected to the vehicle		
18.3	Diagnostic access (e.g. dongles in OBD port) used to facilitate an attack, e.g. manipulate vehicle parameters (directly or indirectly)	M22	Security controls shall be applied to external interfaces

*Figure 2-2: Excerpt of the assignment of threats to mitigations of UN R155 [**UN21**].*

sioning of road vehicle electrical and electronic (E/E) systems, including their components and interfaces. It defines a framework (cf. Figure 2-3) that includes requirements for cybersecurity processes and a common language for cybersecurity risk communication and management.

The focus of my work is the concept phase, for which the following sections are relevant: Section 9 (Concept) covers requirements for activities that determine cybersecurity risks, cybersecurity goals and cybersecurity requirements for an item. For the determination of these points, reference is made to Section 15. Section 15 (Threat analysis and risk assessment methods) includes modular methods for analysis and assessment to determine the extent of cybersecurity risk so that treatment can be pursued.

Fundamental for ISO/SAE 21434 and especially for the concept phase is the model shown in Figure 2-4. This model represents the relationships between the item, functions, components and terms. This model content is further detailed in the individual sections of ISO/SAE 21434 in the form of requirements.

ISO/SAE 21434 requires the creation of a total of 15 work products as part of the concept phase (cf. Figure 2-5). A TARA has to be conducted to determine cybersecurity goals [WP-09-02]. Thereby, the TARA consists of several individual work products ([WP-15-01]-[WP-15-08]). The main result of the concept phase is the cybersecurity concept [WP-09-06]. The cybersecurity concept consists of cybersecurity requirements derived from the cybersecurity goals and a comprehensive view of the item.

ISO/SAE 21434 describes requirements for the creation of work products. It does not define how the work products can be created.

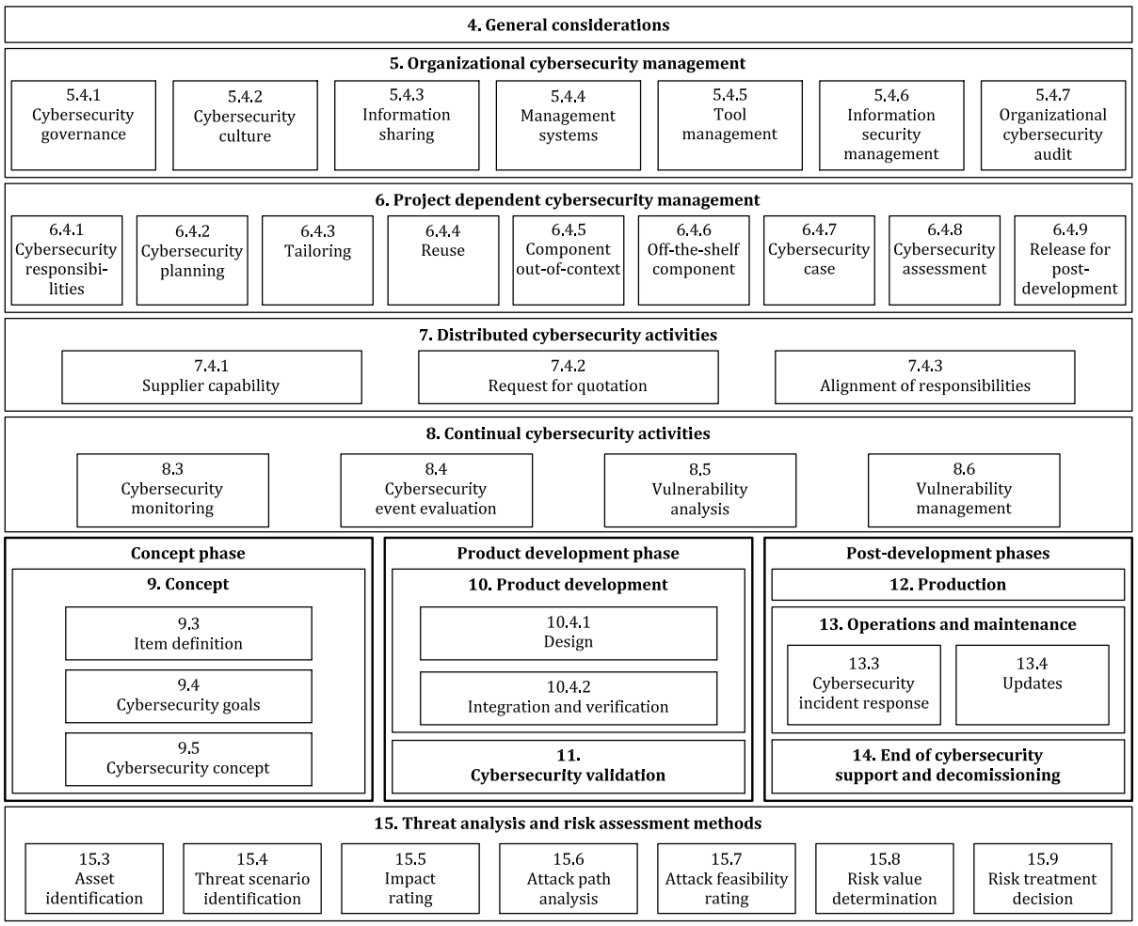


*Figure 2-3: Overview of ISO/SAE 21434: Road Vehicles - Cybersecurity Engineering [**ISO21**].*

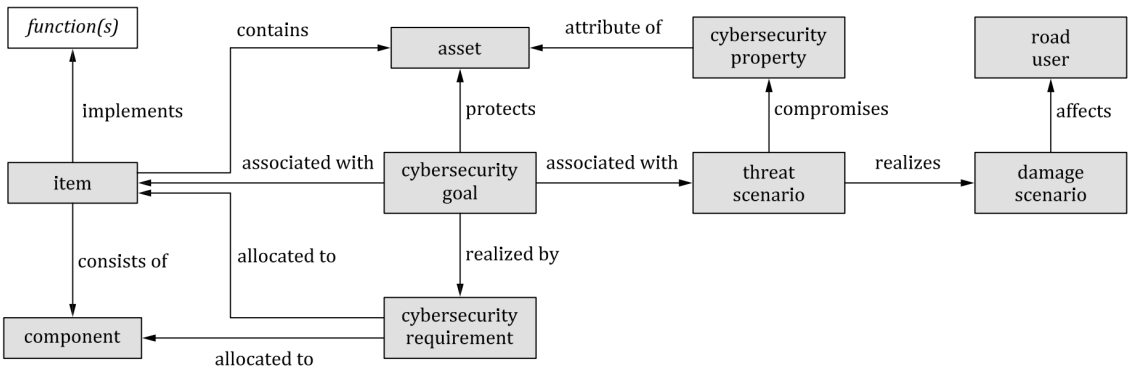


*Figure 2-4: Relationships between the item, functions, components and terms of ISO/SAE 21434 [**ISO21**].*

Concept phase	
9.3 Item definition	[WP-09-01] Item definition
9.4 Cybersecurity goals	[WP-09-02] TARA
	[WP-09-03] Cybersecurity goals
	[WP-09-04] Cybersecurity claims
	[WP-09-05] Verification report for cybersecurity goals
9.5 Cybersecurity concept	[WP-09-06] Cybersecurity concept
	[WP-09-07] Verification report of cybersecurity concept
Threat analysis and risk assessment methods	
15.3 Asset identification	[WP-15-01] Damage scenarios
	[WP-15-02] Assets with cybersecurity properties
15.4 Threat scenario identification	[WP-15-03] Threat scenarios
15.5 Impact rating	[WP-15-04] Impact ratings with associated impact categories
15.6 Attack path analysis	[WP-15-05] Attack paths
15.7 Attack feasibility rating	[WP-15-06] Attack feasibility ratings
15.8 Risk value determination	[WP-15-07] Risk values
15.9 Risk treatment decision	[WP-15-08] Risk treatment decisions

*Figure 2-5: Activities and work products of ISO/SAE 21434 relevant to this work. [**ISO21**].*

2.2 Model-Based Systems Engineering

This section is based on [**GDE+19**].

Model-Based Systems Engineering (MBSE) refers to the concept of a continuous description and analysis of the system to be developed on the basis of models, from the early phase of conception through the entire product life cycle. The models describe the system to be developed from different perspectives. Each viewpoint is an aspect of the system, such as structure or behaviour. For each aspect, a separate model can be created, which is called a partial model. The sum of all partial models with the links between the model elements is called the system model.

The system model is a coherent set of partial models that is created in the course of requirements definition and conceptualisation. In the early phase of product development, documents are mostly used. The majority of the system specification is only available in text form. The disadvantage is that this type of systems engineering leaves a lot of room for interpretation and the documents are difficult to maintain. Furthermore, this often leads to inconsistencies that are not immediately apparent. If a change is made in the course of development or the operational phase, this must be taken into account in the specification. The effects of such changes on other documents cannot be fully described in the text and are then often overlooked.

With the current methods and tools from systems engineering, time-consuming and cost-intensive coordination between experts from the disciplines involved occurs repeatedly during development. One main cause is the lack of methods and tools to promote interdisciplinary thinking and action in systems development. A multidisciplinary modelling approach is needed that closes the gap between the requirements and the established

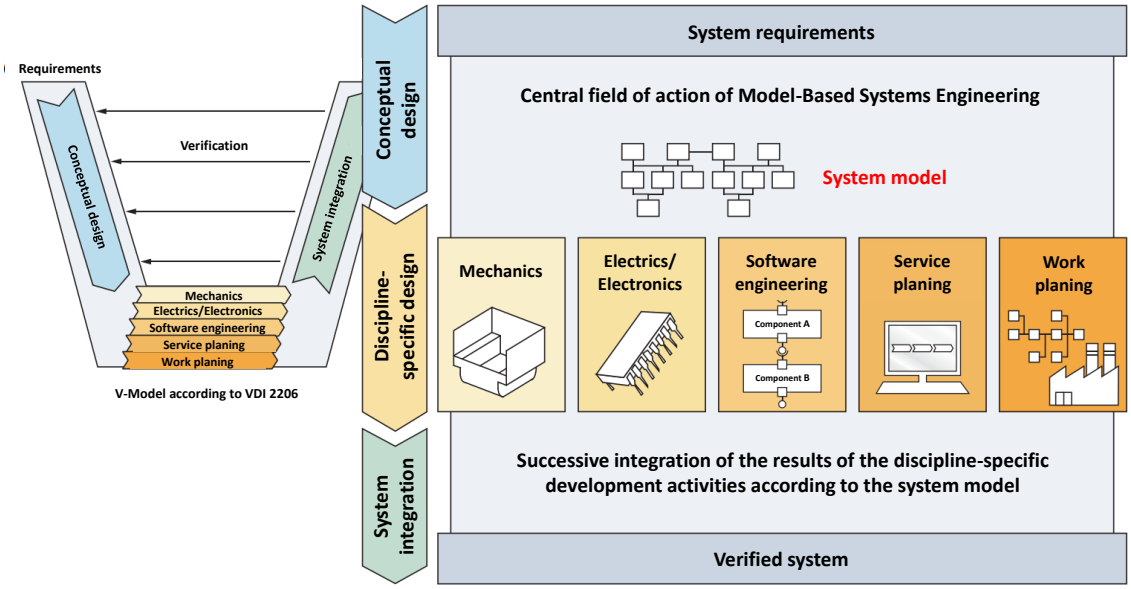


*Figure 2-6: The system model as a basis for closing the gap between system requirements and the discipline-specific development activities [***GDE+19***].*

methods of the individual disciplines (cf. Figure 2-6).

The corresponding modelling technique must therefore capture all relevant aspects of the system to be developed and represent them as partial models, as well as depict the mutual relationships between the partial models. This is where the MBSE approach comes in. With the system model, a model is created right from the start that contains the requirements and the system specification and, based on this, enables initial analyses at an early stage. The system model is the starting point for the discipline-specific design. It is the basis for communication and cooperation between the experts involved throughout the entire product development process and serves as a platform for maintaining the consistency of all partial models created in the course of this process.

2.2.1 Advantages of MBSE

The step from a document-centred to a model-based approach offers several advantages:

- Interdisciplinary system approach: The holistic description of the system contributes to a uniform understanding of all disciplines involved right from the start. This is a fundamental prerequisite for the goal-oriented development of a complex multidisciplinary system.
- Transparent representation of the system interdependencies: By representing the interactions of the design aspects, the subject-specific models of the individual disciplines are integrated via the system model. This creates a basis for ensuring the consistency of the sub-models throughout the entire product development process. For example, all aspects are linked to the requirements in the system model, and the

effects of changes in a sub-model can be traced back.

- Transition to the individual disciplines: The system model is created at the beginning of the development. It forms the starting point for the concretisation of the system from the perspective of the respective disciplines. In the course of this concretisation, the system model is updated and refined if necessary.
- Verification and validation: During development, it must be ensured that the actual system properties match the required properties and that the system functions reliably in all operating situations in the target environment. The system model, which is created at an early stage and continuously updated, creates the prerequisite for deriving test scenarios to validate the system properties during development.
- Coordination of the development process: The system model has the potential to harmonise system design and project management. This provides a powerful platform for the management of development activities. The system model enables a holistic and interdisciplinary view of the system.

2.2.2 Components for describing the system model

A model has a certain degree of formalisation and can thus be represented by a computer. To describe a system model, a graphical modelling language, a modelling method and a modelling tool are needed. Only a properly matched combination of language, method and tool enables effective and efficient use of system modelling in a company. The modelling language, considered in isolation, is only a means of expression. How and for what purpose this language is used is determined by a method. This method specifies what must be specified and the order in which the information is produced. Figure 2-7 shows examples of modelling languages, methods and software tools for use in MBSE.

2.2.3 CONSENS

This section is based on [**GDE+19**].

The CONSENS specification technique consists of a modelling language with different partial models and a modelling method.

The system to be developed is described by seven partial models (cf. Figure 2-8): Environment Model, Application Scenarios, Requirements, Functional Hierarchy, Active Structure, Shape and Behavioural Model. For each of these partial models, specific model constructs exist that allow a generally understandable specification of the system. system.

The partial models Environment Model and Active Structure have the same modeling elements. The partial model Environment defines the system boundary and describes the interaction of the system with its environment. A distinction is made between the relationship types substance, energy, mechanical connection, information and measurement

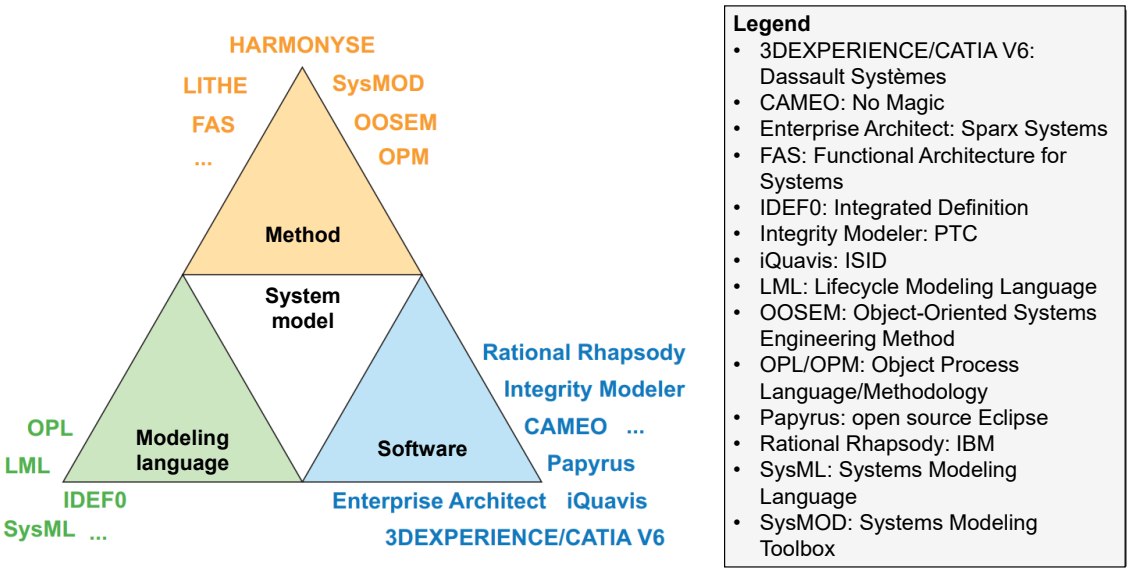


*Figure 2-7: The basis for describing a system model is a combination of graphical modelling language, modelling method and modelling software [**GDE+19**].*

information. The Environment Model treats the system as a black box. The white-box view takes place in the partial model of the Active Structure. This describes the system elements that make up the system and their interactions, also with the help of flow relationships and logical connections.

The CONSENS modelling method describes an iterative procedure for developing the system model of mechatronic systems. This addresses seven aspects. These aspects - and thus the corresponding partial models - are created in an interplay. In the first step, the system boundaries are defined with the partial model Environment Model. In parallel, different situations and the desired behaviour of the system are described in Application Scenarios. Requirements are then derived on the basis of these two partial models. The analysis of the requirements leads to Functions, which are described as noun-verb constructs and ordered in a Function Hierarchy. To fulfil the functions, solution patterns are determined and transferred into system elements. These are synthesised into the Active Structure, which describes the basic architecture of the system and the relationships. Based on this, the behaviour and the shape are subsequently modelled.

2.2.4 Effect Chain Modeling Language

This section is based on [**Sch20**].

Effect chain modelling consists of a modelling method and a modelling language. The graphical elements of the Effect Chain Modeling Language (ECML) are shown in Figure 2-9. Effect chain modelling is oriented towards the levels of the V-model and begins with the representation of the system in its environment. Based on this, further levels are

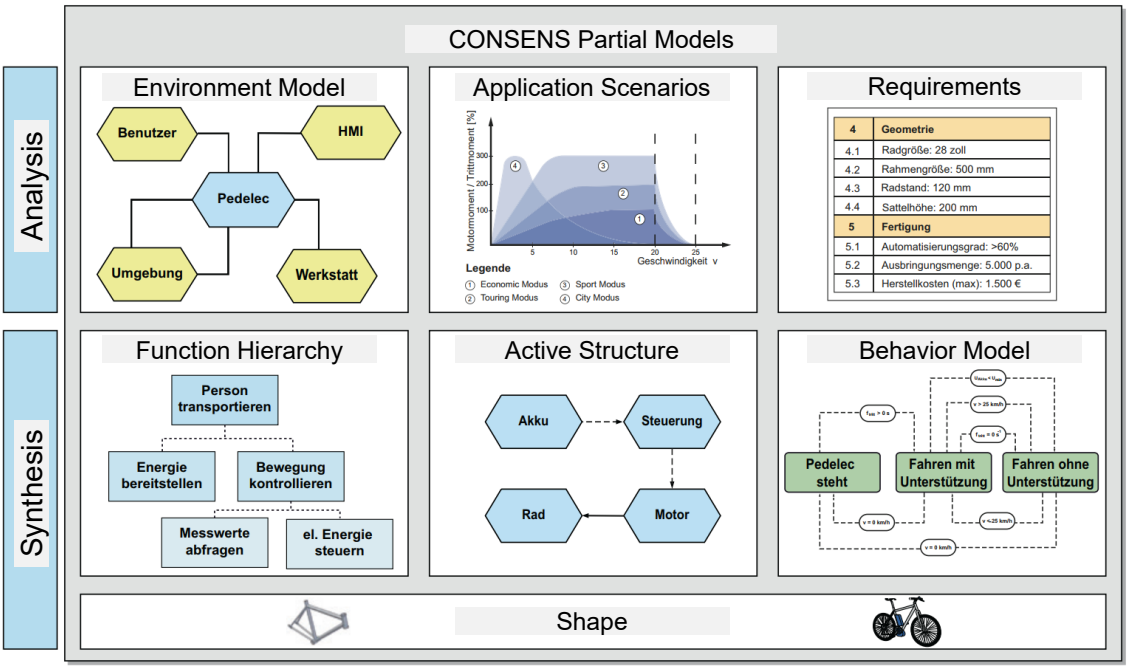


Figure 2-8: Partial models of CONSENS.

modelled. The individual systems and components are interconnected via interactions and interfaces.

The effect chain model gives an overall view of the interactions and interfaces in a system. This includes the modelling of interactions and interfaces with the environment and intentional and unintentional influences between the components in the overall system context. The following flow types between two elements (e.g. components) are distinguished: Mechanical, Information/Software, Substance/Material, Electrics/Electronics, Power/High voltage, Thermal energy, Light/Optics, Airborne sound. With regard to interactions, three attributes are used: Intentional, Unintentional and Misuse.

Effect chain modelling is used to visualise and detail the structure of a mechatronic system. Functional aspects are not part of the effect chain modelling. Due to the few modelling elements, effect chain modelling is easy to learn and use.

2.2.5 Systems Modeling Language

This section is based on [**OMG23-ol**; **Nom23-ol**].

The OMG Systems Modelling Language (SysML) is a general-purpose graphical modelling language for the specification, analysis, design and verification of complex systems that may include hardware, software, information, personnel, procedures and facilities. In particular, the language provides graphical representations with a semantic basis for modelling system requirements, behaviour, structure and parameters, which is used for integration with other engineering analysis models. It represents a subset of UML 2 with extensions necessary to meet the requirements for systems engineering.

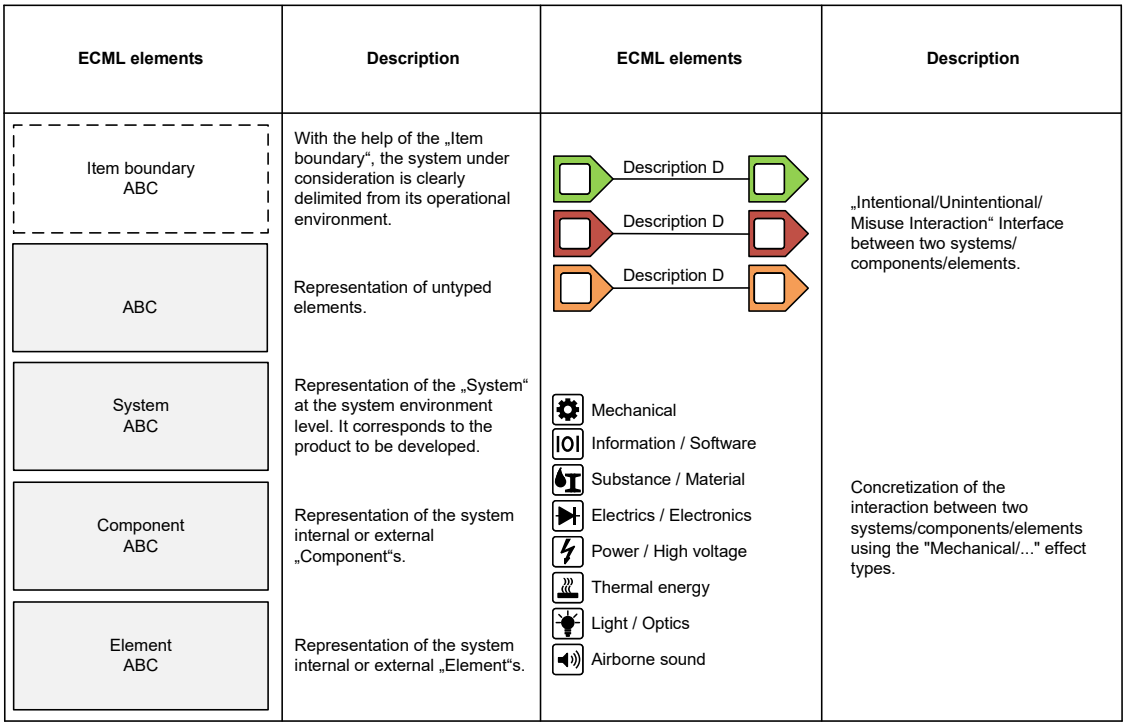


Figure 2-9: Graphical elements of the Effect Chain Modeling Language (ECML).

The diagrams of SysML are shown in Figure 2-10. The Package Diagram reflects the organisation of a model. The Requirements Diagram represents text-based requirements and their relationship to other requirements, structure and verification elements. The Requirements Diagram depicts text-based requirements and their relationship to other requirements, structure and verification elements to support traceability of requirements. Structure diagrams are used to represent static aspects of the system. Typically, they are used for modelling the system architecture. The Parameter Diagram represents constraints on property values used to support technical analysis. Behaviour diagrams can be used to describe system behaviour.

In the following sections I describe the diagrams relevant to my work. I limit the description of these diagrams to the constructs relevant to my work.

2.2.5.1 Use Case Diagram

The purpose of a Use Case Diagram (cf. Figure 2-11) is to provide a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases) and any dependencies between those use cases.

A Use Case Diagram describes the usage of a system. The associations between actors and use cases represent the communications that occur between actors and subjects to achieve the functionalities associated with the use cases. The subject of a use case can be represented by a system boundary. The use cases enclosed by the system boundary represent the functionalities performed by behaviours (activity diagrams, sequence diagrams and

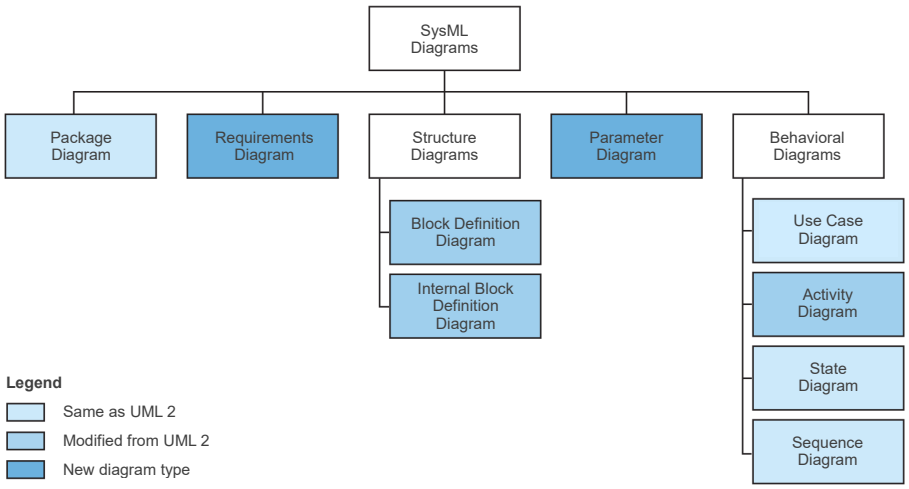


Figure 2-10: Diagrams of the Systems Modeling Language (SysML).

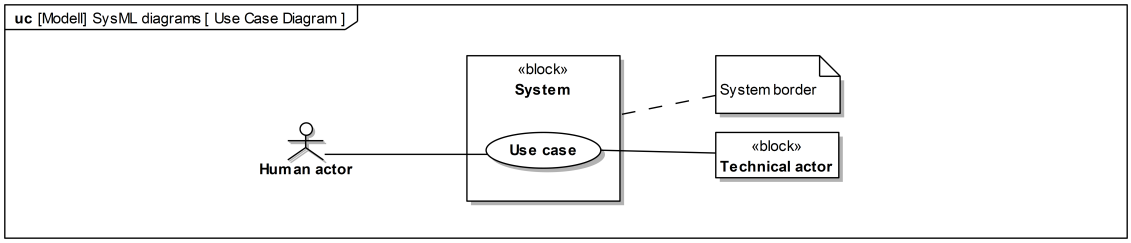


Figure 2-11: SysML Use Case Diagram.

state machine diagrams).

Actors can interact with the system either directly or indirectly. Actors are connected to use cases by communication paths, each represented by a relationship.

2.2.5.2 Block Block Diagram

The Block Definition Diagram (BDD) (cf. Figure 2-12) defines the features of a block and any relationships between blocks, such as associations, generalisations and dependencies, in terms of properties, operations and relationships.

An Interface Block is a special kind of block for typing Proxy Ports. It has no behaviours or internal parts. It usually contains a set of Flow Properties. A Proxy Port is a port that specifies features of owning Blocks or Part Properties that are available to external Blocks through external Connectors to the ports. It can only be typed by an Interface Block. Ports can be conjugated. A conjugated port inverts the flow direction of all flows of a port.

2.2.5.3 Internal Block Diagram

An Internal Block Diagram (IBD) (cf. Figure 2-13) captures the internal structure of a Block in terms of properties and connections between properties. A Block contains properties so that its values, parts and references to other blocks can be specified. However,

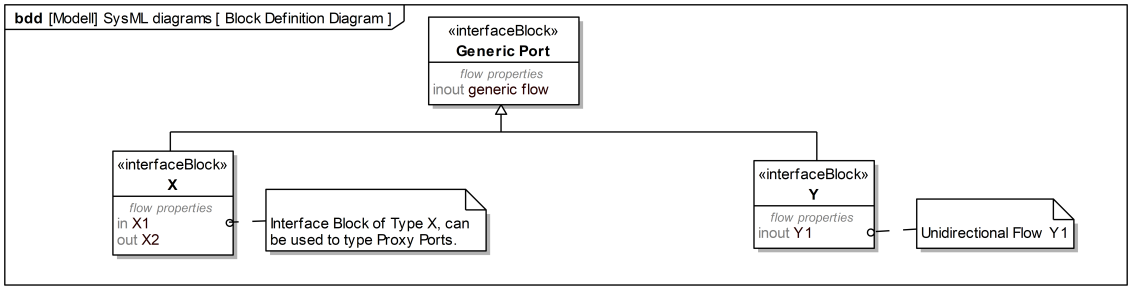


Figure 2-12: SysML Block Definition Diagram (BDD).

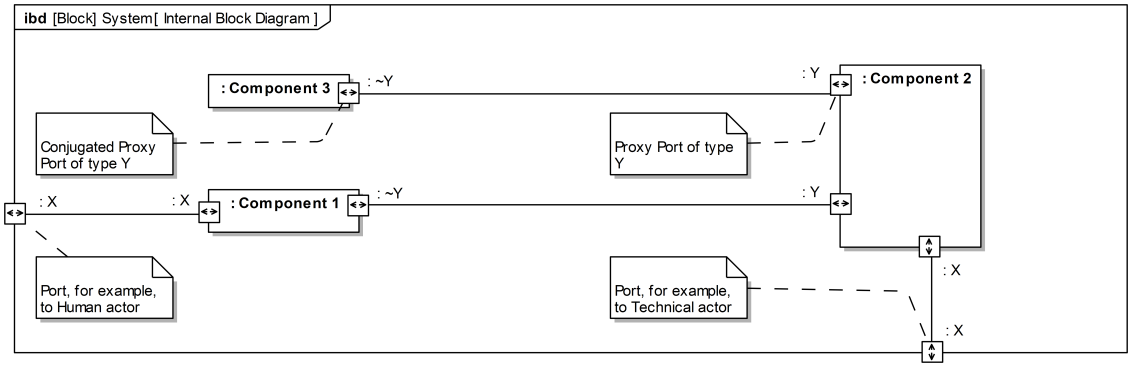


Figure 2-13: SysML Internal Block Diagram (IBD).

while an IBD created for a Block (as an inner element) will only show the inner elements of a classifier (parts, ports and connectors), an IBD created for a package will show additional elements (shapes, notes and comments).

All properties and connectors that appear within an IBD belong to a Block, whose name is written in the diagram heading. This Block is the context of the diagram.

2.2.5.4 Sequence Diagram

The Sequence Diagram (cf. Figure 2-14) is a behavioural diagram that focuses on the exchange of messages between different lifelines. A lifeline represents a single participant in the interaction. A Sequence Diagram shows the interaction information with a focus on the temporal sequence. The diagram has two dimensions: the vertical axis representing time and the horizontal axis representing the objects involved.

2.2.5.5 Requirements Diagram

The Requirements Diagram (cf. Figure 2-15) is useful for showing traceability from the requirements to the dependent elements in the system model. This diagram provides modelling constructs to represent text-based requirements and relate them to other modelling elements. These requirement modelling constructs are used to bridge requirements between requirements management tools and other SysML models.

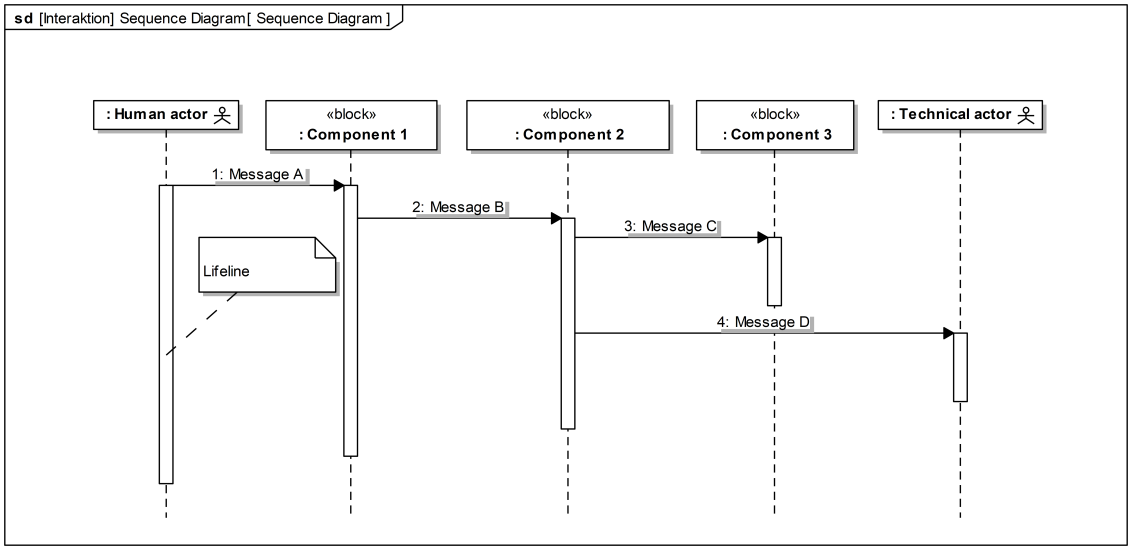


Figure 2-14: SysML Sequence Diagram.

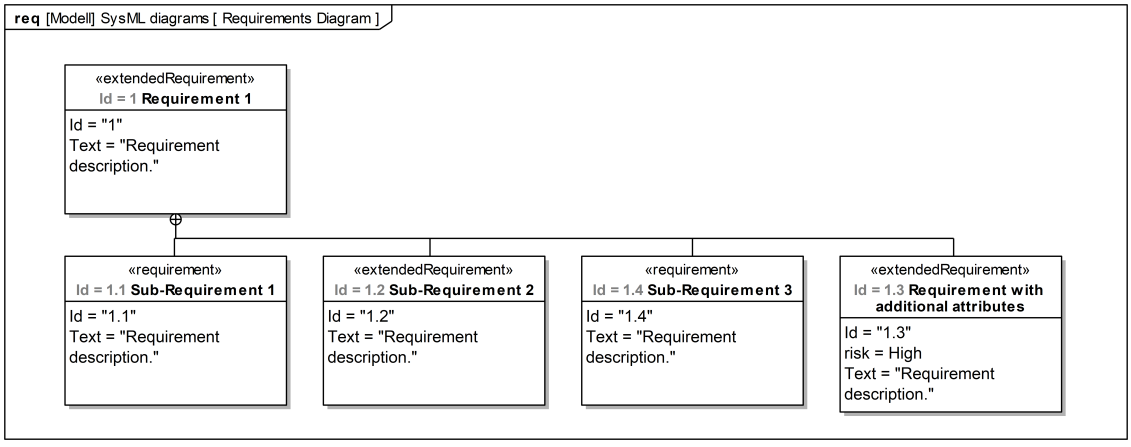


Figure 2-15: SysML Requirements Diagram.

#	Name	Text	Traced To	Risk
1	⊟ 1 Requirement 1	Requirement description.	Use case System	
2	1.1 Sub-Requirement 1	Requirement description.	Component 1	
3	1.2 Sub-Requirement 2	Requirement description.	Component 2	
4	1.4 Sub-Requirement 3	Requirement description.	Component 3	
5	1.3 Requirement with additional attributes	Requirement description.	Proxy Port X Proxy Port Y	High

Figure 2-16: SysML Requirements Table.

2.2.5.6 Requirements Table

A Requirement Table (cf. Figure 2-16) is used to collect requirements in a table. Each row in the table represents a requirement. Each requirement has the attributes Id, Name and Text. In the table, the attributes are represented as columns. Extended Requirements can be used to create specialised requirements with additional attributes.

2.3 Domain-specific approaches

2.3.1 Determination of the Automotive Safety Integrity Level

This section is based on [**GDE+19**].

ISO 26262 [**ISO18**] must be taken into account by car manufacturers in accordance with the German Product Liability Act. ISO 26262 provides guidelines and recommendations, in particular for the determination of risk levels for systems and components.

One part of ISO 26262 describes the Automotive Safety Integrity Level (ASIL) (cf. Figure 2-17). An ASIL represents the degree of danger/impact on affected persons in the event of a failure. For this purpose, the ASIL assessment of a technical system is composed of the probability being exposed to a fault (Exposure), the controllability and the severity of the fault (Severity). The ASIL (level) results from the assessment of the three criteria. The spectrum ranges from ASIL A (marginal, e.g. time delay of the image of a rear view camera) to ASIL D (catastrophic, e.g. sudden triggering of the airbag). Depending on the level, appropriate measures are required to prevent the risk and to prove the avoidance of the risk. For failures rated as non-critical (QM), the application of normal procedures according to quality management is sufficient.

2.3.2 Fault Tree Analysis

This section is based on [**ESH15**].

Fault Trees represent the relationship between events and their linkage using Logic Gates. The structure of a Fault Tree corresponds to a Tree. The modelling is done according to certain rules. The basic structure and layout is illustrated by an example in Figure 2-18.

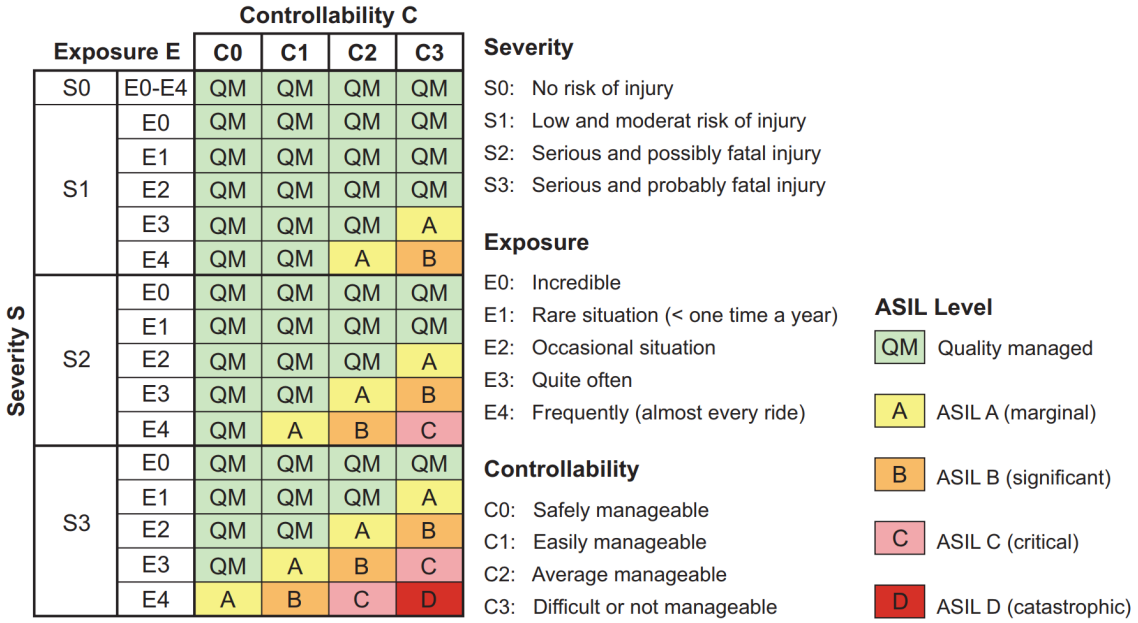


*Figure 2-17: Determination of the Automotive Safety Integrity Level (ASIL) [**GDE+19**].*

An event describes the occurrence of a system state, which usually characterises a fault or error state. In principle, events that are not errors can also be expressed. All events are noted as rectangles in which a short text describes the event. Events can be further detailed by various gates. This refinement step corresponds to a specification in the sense of a decomposition into individual partial events that are logically linked.

The example in Figure 2-18 shows several Events, two Gates and one Transfer Gate. The root of the Fault Tree represents the main event of the analysis. Below this is an OR Gate with two input events E1 and E2. This expresses that the Top Level Event can occur either through E1 or through E2. While the tree on the left ends at E1, Event E2 has been further subdivided by means of an AND Gate, so that Event E2 has been specified as E3 and a Transfer Gate. Transfer Gates can be used to structure a Fault Tree into manageable sub-trees. Transfer Gates are represented with the help of a triangle. The three events E1, E3 and E4 are called Primary Events because they are not subdivided further. Event E2 is called an Intermediate Event. Elementary events that cannot be further subdivided are called Basic Events and are marked with a circle. Events that are not investigated further (Undeveloped Events) are marked with a diamond.

2.3.3 Attack Potential-Based Approach

This section is based on [**ISO21**].

Attack potential is defined as a measure of the effort required to attack an item or component, expressed in terms of an attacker's expertise and resources. Attack potential is based on five core parameters: expertise, knowledge of the item or component, equipment, window of opportunity and elapsed time. Figure 2-19 presents an example configuration

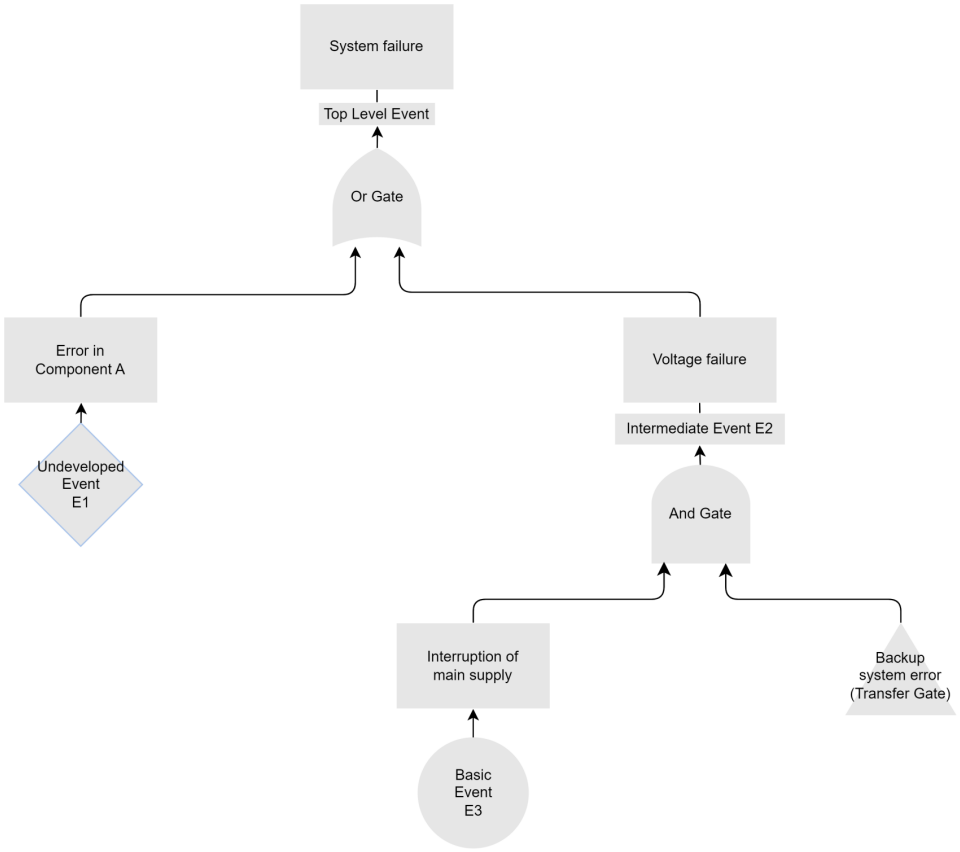


*Figure 2-18: Example of a Fault Tree [***ESH15***].*

of the core parameters.

Expertise refers to the attacker's abilities, relative to their skill and experience. Item or component parameter knowledge is related to the amount of information the attacker has about the item or component. The equipment parameter refers to the tools available to the attacker to discover the vulnerability and/or execute the attack. The window of opportunity parameter refers to the access conditions (time, type) for a successful attack. It combines the type of access (e.g. logical and physical) and the duration of access (e.g. unlimited and limited). Depending on the type of attack, this can include the discovery of possible targets, access to a target, exploit work on the target, time to execute an attack on a target, remain undiscovered, bypass detection and cybersecurity controls, etc. The elapsed time parameter includes the time it takes to identify a vulnerability and to develop and (successfully) apply an exploit. Therefore, this rating is based on the state of expert knowledge at the time of the rating. Numerical values can be defined for each parameter.

The attack feasibility rating is carried out as follows (cf. Figure 2-19): For each attack, a proficiency level must be determined for each core parameter. There is a numerical value for each proficiency level. The numerical values of the selected proficiency levels are added up. The added value belongs to an interval called the Attack Feasibility Level (ATF). The ATF is used together with the (safety) impact level to determine the risk of an attack.

	Value	Description
Specialist expertise	0	**Layman**: Unknowledgeable compared to experts or competent persons, with no particular expertise.
	3	**Proficient**: Knowledgeable in the sense that they are familiar with the security behaviour of the type of product or system.
	6	**Expert**: Knowledge of the underlying algorithms, protocols, hardware, structures, security behaviour, security principles and concepts used, techniques and tools for defining new attacks, cryptography, classic attacks for the product type, attack methods, etc. implemented in the product or system type.
	8	**Multiple experts**: At the expert level, different areas of expertise are required for different steps of an attack.
Knowledge of the item or component	0	**Public information:** Publicly available information on the item or component (e.g. as obtained from the Internet).
	3	**Restricted information:** Restricted information about the item or component (e.g. knowledge controlled within the development organisation and shared with other organisations under a non-disclosure agreement).
	7	**Confidential information:** Confidential information about the item or component (e.g. knowledge shared between discrete teams within the developer organisation and access to which is restricted to members of the specified teams only).
	11	**Strictly confidential information:** Strictly confidential information about the item or component (e.g. knowledge known only to a small number of people, access to which is strictly controlled on a strict need to know basis and individual undertaking).
Equipment	0	**Standard:** Equipment is readily available to the attacker. This equipment can be part of the product itself (e.g. a debugger in an operating system) or can be easily obtained (e.g. internet sources, protocol analysers or simple attack scripts).
	4	**Specialized:** Equipment is not readily available to the attacker, but can be acquired without undue effort. This may involve the purchase of moderate amounts of equipment (e.g. use of hundreds of PCs connected via the Internet would fall into this category) or the development of more sophisticated attack scripts or programs. If clearly different test beds with specialised equipment are required for distinct steps of an attack, this would be rated as bespoke.
	7	**Bespoke:** Equipment is specially produced (e.g. very sophisticated software) and not readily available to the public (e.g. black market), or the equipment is so specialised that its distribution is controlled, possibly even restricted. Or the equipment is very expensive.
	9	**Multiple bespoke:** Introduced to allow for the situation where different customised equipment may be required for different stages of attacking.
Window of opportunity	0	**Unlimited**: High availability over public/untrusted network with no time limit (i.e. asset is always accessible). Remote access without physical presence or time constraints and unlimited physical access to the item or component.
	1	**Easy**: High availability and limited access time. Remote access without physical presence at the item or component.
	4	**Moderate**: Low availability of the item or component. Restricted physical and/or logical access. Physical access to the interior or exterior of the vehicle without the use of special tools.
	10	**Difficult**: Very low availability of the item or component. Impractical level of access to the item or component to execute the attack.
Elapsed time	0	**<= 1 day**
	1	**<= 1 week**
	4	**<= 1 month**
	17	**<= 6 months**
	19	**> 6 months**

	Sum	Description
Attack feasibility rating	0-13	**High (Values between 0 - 9)**
	14-19	**Medium**
	20-24	**Low**
	>= 25	**Very low**

*Figure 2-19: Exemplary configuration of the core parameters of the Attack Potential-based Approach [**ISO21**].*

	Description	Methods to provide confidence that cybersecurity activities are performed with apropriate rigour	Independence scheme to provide confidence that the cybersecurity activities performed are appropriate
CAL1	Low to moderate cybersecurity assurance is required	Requirement based testing	Not needed
CAL2	Moderate cybersecurity assurance is required		Cybersecurity assessments are carried out by a different person than the originator
CAL3	Moderate to high cybersecurity assurance is required	All interactions between components are tested	Cybersecurity assessments are carried out by a person in a different team than the originator
CAL4	High cybersecurity assurance is required	All combinations of interactions between components are tested	Cybersecurity assessments are carried out by a person who is independent regarding management, resources and release authority from the originating department

*Figure 2-20: Example of CALs and the expected level of rigour in cybersecurity assurance measures [***ISO21***].*

2.3.4 Cybersecurity Assurance Level

This section is based on [**ISO21**].

The Cybersecurity Assurance Level (CAL) classification scheme is used to specify and communicate a set of assurance requirements, in terms of levels of rigour, to provide confidence that the protection of the assets of an item or component is adequately developed. This CAL classification scheme does not specify technical requirements for cybersecurity controls, but is used to drive cybersecurity engineering and provide a common language for communicating cybersecurity assurance requirements between participating organisations.

A CAL classification scheme can be used to determine the level of rigour with which cybersecurity activities are performed, in terms of the effort required to provide the required assurance. A CAL can be used to select development and verification methods, vulnerability identification and analysis methods, and cybersecurity assessment approaches.

Figure 2-20 provides an example of a set of CALs and guidance on their use during the concept and product development phases. For each increase in CAL, the corresponding methods represent a meaningful increase in the assurance of the item or component through design, verification and cybersecurity assessment.

2.4 Problem delimitation

In summary, the following can be stated: UN Regulation No. 155 (UN R155) (cf. Section 2.1.1) describes uniform conditions for the approval of vehicles with regard to cybersecurity and the implementation of a cybersecurity management system (CSMS). Based on an established and approved CSMS, OEMs must provide vehicle type-specific evidence that identifies appropriate measures to mitigate the cyber risks associated with their products

(e.g. consistent risk assessment, relevant mitigation measures, etc.). OEMs must define relevant cybersecurity requirements to pass on to their suppliers to ensure end-to-end security throughout the supply chain.

UN R155 explicitly requires that manufacturers have qualified personnel with cybersecurity competence and expertise in automotive risk assessment. Here, reference is made to ISO/SAE 21434 (cf. Section 2.1.2). While UN R155 focuses on type approval, ISO/SAE 21434 describes concrete requirements for a process framework to ensure cybersecurity in the automotive sector. The implementation of the cybersecurity process framework in the company represents the CSMS required by UN R155.

ISO/SAE 21434 specifies technical requirements for the management of cybersecurity risks related to the design, product development, production, operation, maintenance and decommissioning of road vehicle electrical and electronic (E/E) systems, including their components and interfaces. It defines a framework that includes requirements for cybersecurity processes and a common language for communicating and managing cybersecurity risks.

Model-Based Systems Engineering (MBSE) (cf. Section 2.2) is an interdisciplinary approach to the holistic description of complex, intelligent and networked systems. In this approach, the information about a system to be developed is no longer based exclusively on documents, but primarily on models. With the help of models, complexity can be reduced, e.g. by using different models with different levels of detail for different perspectives and system levels. Overall, these models and their interrelationships form the system model.

Regarding the framework to be developed, it is important to highlight the following challenges:

Lack of a holistic approach to the creation of a cybersecurity concept UN R155 will become mandatory in the European Union from July 2022 for all new vehicle types and from July 2024 for all newly produced vehicles. This means that vehicles developed without a valid CSMS cannot be approved in the European Union.

In particular, UN R155 requires OEMs to have qualified personnel with cybersecurity competence and expertise in vehicle risk assessment. In this context, explicit reference is made to ISO/SAE 21434.

ISO/SAE 21434 describes concrete requirements for a process framework to ensure cybersecurity in the automotive sector. The implementation of the cybersecurity process framework in the company represents the CSMS required by UN R155. As part of the risk analysis, critical threat scenarios must be identified and suitable mitigation measures must be selected. To ensure compliance with UN R155, it must be proved that the catalogue of threats and mitigations of UN R155 has been taken into account.

ISO/SAE 21434 describes requirements for activities to determine cybersecurity risks, cybersecurity goals and cybersecurity requirements for a system to be developed as part

of the concept phase. In addition, ISO/SAE 21434 describes requirements for Threat Analysis and Risk Assessment (TARA) methods. The result of the concept phase is the cybersecurity concept.

To create the cybersecurity concept, 15 work products must be created that must fulfil numerous requirements of ISO/SAE 21434. Although ISO/SAE 21434 describes requirements for the creation of the work products, ISO/SAE 21434 does not define how these work products have to be created.

High complexity when creating the cybersecurity concept The development of complex intelligent and connected systems requires the collaboration of different disciplines. One challenge lies in the complex collaboration and communication in the concept phase.

Model-Based Systems Engineering (MBSE) supports the holistic description of complex systems. With the help of models, complexity can be reduced, e.g. by using different models with different levels of detail for different perspectives and system levels. All of these models and their interrelationships form the system model.

A graphical modelling language, a modelling method and modelling software are needed to describe a system model. Only a properly tailored combination of language, method and software enables effective and efficient use of system modelling in a company. Viewed in isolation, the modelling language is only a means of expression. How and for what purpose this language is used is determined by a method. This method determines what must be specified and the order in which the information is created. With the help of modelling software, the models created can be technically managed.

For the development of the cybersecurity concept using MBSE, a suitable modelling method, modelling language and modelling software is required.

Considering the challenges mentioned, there is a need for a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*.

Based on the identified challenges, I describe the resulting fields of action of my work in the following (cf. Figure 2-21):

Field of action 1 (Method): A concrete approach is required for the creation of the 15 work products as part of the concept phase, which must comply with the numerous requirements of ISO/SAE 21434. The result of this procedure is the cybersecurity concept.

The creation of the cybersecurity concept requires in particular the creation of the item definition, carrying out the Threat Analysis and Risk Assessment (TARA) and the derivation of the cybersecurity goals and cybersecurity requirements.

In the context of risk analysis, ISO/SAE 21434 refers to existing approaches that have to be used. To determine the safety impact of an attack, the ASIL risk classification scheme established in the automotive sector (cf. Section 2-17) serves as a reference.

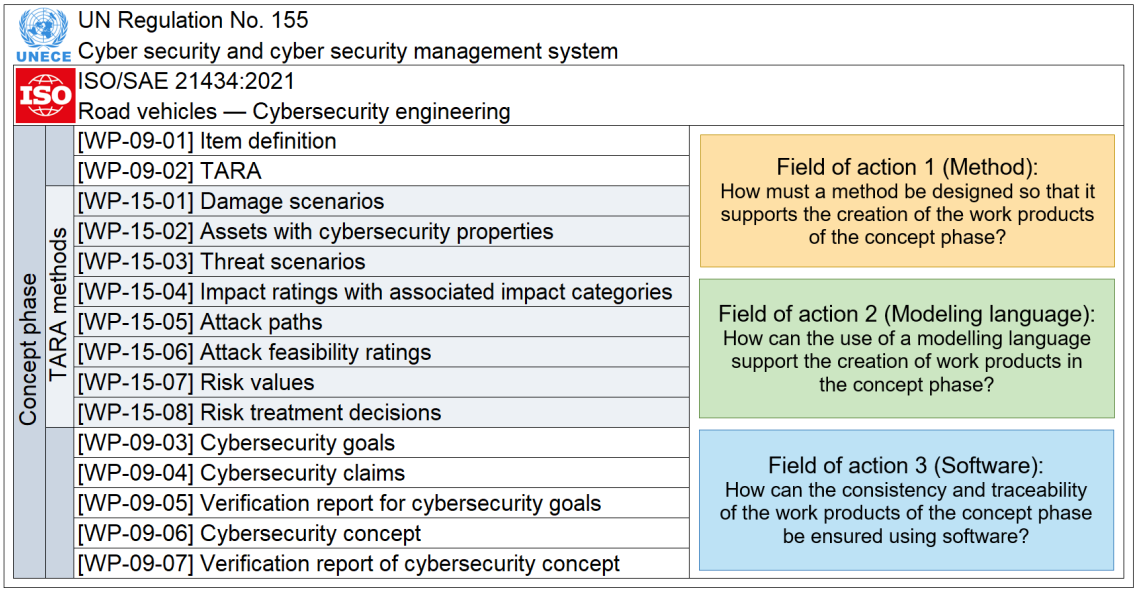


Figure 2-21: Overview of the fields of action for the solution to be developed.

For assessing the feasibility of attacks, reference is made in particular to the Attack-Potential-Based Approach (cf. Section 2-19). For the modelling of attack paths, reference is made to the use of attack trees. Fault trees, which add logical gates to (attack) trees (cf. Section 2-18), are suitable for the traceability of the cause of the attack and its effect on the system to be developed.

To facilitate communication between different business units or organisations, reference is made to the use of Cybersecurity Assurance Levels (CAL) (cf. Section 2-20). By using CALs, it can be determined, depending on the importance of a system to be developed, at which effort a TARA must be carried out.

Field of action 2 (Modeling Language) A suitable modelling language is needed to create the 15 work products as part of the concept phase.

The collaboration in the concept phase often takes place in workshops with several participants from different disciplines. Often these are leading experts who have an overview over several projects. In addition, such experts have limited time. This means that knowledge extraction has to be done in a short time and complex components of a modelling language cannot be used. [**Jap20**; **Jap21**].

For extracting and modelling expert knowledge, simple modelling languages such as ECML (cf. Section 2.2.4) and CONSENS (cf. Section 2.2.3) are suitable. To ensure that such models can also be used in the context of detailed system design, a mapping to modelling languages such as SysML (cf. Section 2.2.5) is necessary.

Field of action 3 (Software) During the concept phase, 15 work products must be created. For each work product, numerous requirements must be fulfilled in order to conform to ISO/SAE 21434. This requires the creation of numerous interconnected models. Modelling software is required to reduce the engineering effort to comply with ISO/SAE 21434. The

use of modelling software enables traceability between work products and change tracking.

2.5 Thesis requirements

This section is based on three scientific papers I have written [**Jap20**; **Jap21**; **JKA+23**].

The problem delimitation in Section 2.4 results in the following requirements for a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*.

R1) Support in the creation of the work products of the concept phase of ISO/SAE 21434: To ensure conformity with ISO/SAE 21434, the required method must support the creation of certain work products of the concept phase. This includes the Item Definition (cf. figure 2-21 [WP-09-01]) and the TARA [WP-09-01] including the associated work products [WP-15-01] - [WP-15-08]. In order to ensure the conformity of ISO/SAE 21434 with UNECE R155, the required method in the context of TARA must take into account the threats and mitigations catalogued in UNECE R155.

The required method must be applicable in (online) workshops. This is because the concept phase is characterised by the collaboration of multiple experts from different disciplines, departments and possibly from several companies and from different locations. The collaboration often takes place in workshops. As a result of increasing digitalisation, these workshops can take place online, independent of any concrete location.

R2) Systematic approach for deriving requirements: To ensure full conformity with the concept phase of ISO/SAE 21434, the required method must also support the derivation of cybersecurity goals and cybersecurity requirements (cf. Figure 2-21 [WP-09-03] - [WP-09-07]). The models created for the Item Definition [WP-09-01] and the TARA [WP-09-02] provide the basis for this.

The Item Definition and the TARA is not sufficient as a communication medium outside of workshops. In particular, communication between car manufacturers and suppliers requires the definition of cybersecurity goals and cybersecurity requirements. The knowledge of the stakeholders gained through the creation of the Item Definition and the TARA can be made more precise through the definition of cybersecurity goals and cybersecurity requirements. In addition, assumed facts in the models can be double-checked by writing down requirements, thus improving the quality of the Item Definition and the TARA.

R3) Use of a standardized modeling language from the fiel of systems engineering: To simplify communication between the stakeholders, but also to transfer the results to other companies, e.g. automotive suppliers, the required method must be based on a standardised and established modelling language from the field of systems engineering, e.g. SysML.

In this context, models should serve as a means of communication so that ambiguities can already be resolved in the workshops. The participants should be able to complement each other with their individual expertise about individual components, necessary functionalities,

interface knowledge and knowledge about frequently occurring problems.

To ensure the applicability of the method in workshops, it should use simple model constructs. This ensures that an interdisciplinary team of mainly leading stakeholders can apply the model constructs without in-depth modelling knowledge.

R4) Realisation in a professional MBSE tool or based on a professionel MBSE platform: For the creation of the 15 work products within the concept phase, the use of modelling software from the field of MBSE is required. For each work product, numerous requirements must be met in order to comply with ISO/SAE 21434. This requires the creation of numerous interrelated models. The modelling software used should be adapted to conform to ISO/SAE 21434, e.g. by means of a Profile or a Template. This should reduce the manual effort to ensure the conformity of the contents to ISO/SAE 21434. Furthermore, the modelling software should ensure traceability between the work products. Ensuring traceability facilitates the traceability of changes to models and requirements across multiple interconnected work products.

R5) Realistic and continuous example from the automotive domain: The required approach must be validated using a realistic and consistent application example from the automotive sector for all 15 work products of the concept phase.

I would like to avoid the following points: I would like to avoid low acceptance of my approach by using an application example that is as realistic and believable as possible. I would like to avoid an improper risk assessment. An unrealistic example could lead to risks being assessed either too high or too low. This would lead to a wrong prioritisation of development activities, as the real threats and vulnerabilities would not be adequately addressed. I would like to avoid irrelevant results. If the application example is unrealistic, the results cannot be transferred to real scenarios. This can lead to misjudgements in choosing the right security measures. I would like to avoid multiple partial analyses. Multiple partial analyses make it difficult to identify interrelationships in the risk analysis that cover several work products.

3 Research method

My research method is based on an extension of the Design Science Research (DSR) approach. For this, I first describe the DSR approach and present the relation to my work. Then I present an extension of the DSR approach that supports a repeated application of the approach. I use this approach to present in detail how I proceeded in my work.

3.1 Design Science Research

According to [**Hev07**], the DSR approach consists of three interrelated cycles and activities (cf. Figure 3-1). The Relevance Cycle introduces contextual environment requirements into research and puts research artefacts into environmental field testing. The Rigor Cycle provides the underlying theories and methods together with domain experience and expertise from the knowledge base into the research and adds the new knowledge generated by the research to the growing knowledge base. The Design Cycle supports a tighter loop of research activity for the construction and evaluation of design artefacts and processes.

Within the scope of my work, I was able to get to know the requirements of the automotive industry in several industry and research projects for the area of the concept phase. The existing literature from the areas of Model-Based Systems Engineering and Security Engineering served as my knowledge base. In numerous workshops with students and experts from industry and research, I was able to test the partial solutions I had developed. In the context of several scientific papers, I was able to contribute to expanding the existing knowledge base in the field of automotive cybersecurity engineering and model-based systems engineering.

3.2 Design Science Research Process Model

According to [**MGS17**], the Design Science Research Process Model (DSR-PM) [**KV08**] is one of the most frequently cited and accepted methods among design science researchers. The DSR-PM consists of the following five phases that can be iteratively repeated. (1) Problem Awareness can be gained from practical experience or from related disciplines. The outcome of this phase is a Suggestion. (2) The Suggestion is closely related to the problem. The Suggestion is a preliminary draft of the solution. (3) In the Development phase the preliminary draft is implemented. The technique of implementation depends on the artefact. (4) When the artefact has been developed, an Evaluation is required. Based on the Evaluation, a Suggestion or the Development can be refined. (5) The Conclusion is the final phase where the research results and contributions are identified. This includes not only the artefact, but also any additional knowledge gained from the process, design and evaluation. The outcome of this phase is a contribution to the research. The implementation of these phases may reveal new problems that can be addressed in a new DSR cycle.

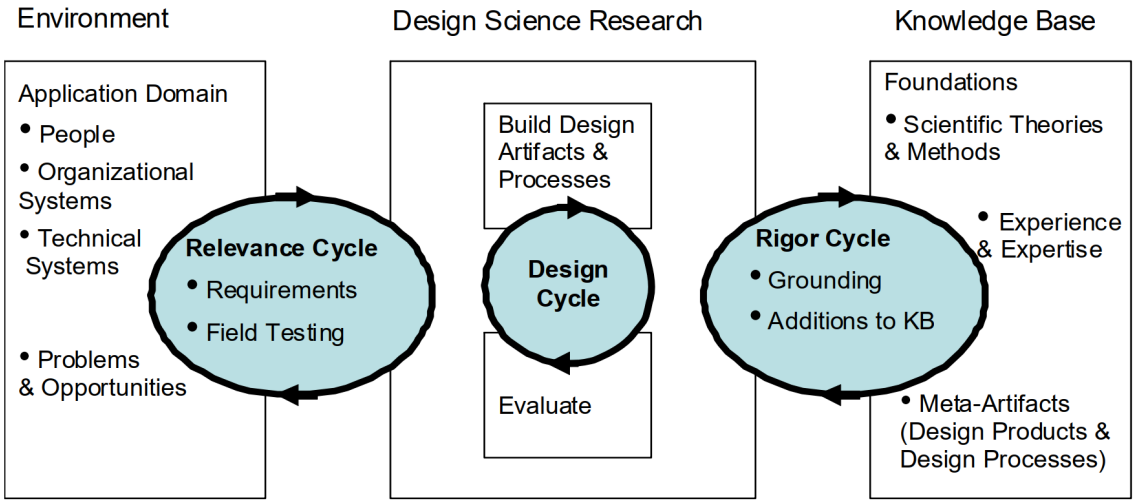


*Figure 3-1: Design Science Research approach [**Hev07**].*

The five phases of the DSR-PM approach can be mapped to the three cycles of the DSR approach as follows. The (1) Problem Awareness phase relates to the Relevance Cycle (Requirements, Field Testing). The (5) Conclusion phase relates to the Rigor Cycle (Grounding, Additions to Knowledge Base). The (2) Suggestion, (3) Development and (4) Evaluation phases refer to the Design Cycle.

In the following, I describe the DSR-PM used in this work with four iterations (cf. Figure 3-2). The evaluation within the four iterations is described in detail in Section 6.

3.2.1 Iteration 1

Awareness of Problem: I started my work by analysing the existing literature on the topic of automotive security engineering.

This included in particular the analysis of SAE J3061 (Cybersecurity Guidebook for Cyber-Physical Vehicle System) [**SAE16**] from 2016, which is well-known in the security engineering field. SAE J3061 is the predecessor of ISO/SAE 21434, which was published in 2021 and on which my work is based. The Attack Potential-Based Approach referenced in SAE J3061 and ISO/SAE 21434, which I present in Section 2.3.3, has thereby become part of my work.

I was able to learn about concrete use cases of security vulnerabilities in the automotive sector in particular by analysing the work of Charlie Miller and Chris Valasek. The security researchers are known for the attack on a Jeep Cherokee from 2015, which they were able to control remotely [**Gre15-ol**]. In [**VM14**], the security researchers describe their preliminary analysis of 21 vehicle architectures and the identification of possible vulnerabilities. This allowed me to learn about security- and safety-relevant components and their relationships to each other in vehicles. In my work, I use a similar level of abstraction for the representation of vehicle architectures as in [**VW14**].

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
Awareness of Problem	Literature analysis on automotive security engineering	Identification of threats is not enough	Requirements engineering as a framework is not sufficient	Selection of solution patterns for the concept phase is sufficient
Suggestions	Integration of the solution idea into requirements engineering	Threat resolution is necessary	Ensure compatibility with ISO/SAE 21434	Improvement of partial results
Development	Extension of existing methods	Development of solution patterns	Continue development according to the activities/ requirements of ISO/SAE 21434	Tailoring the partial results for implementation in workshops with subject matter experts
Evaluation	**Teaching:** workshops with students; theses	**Teaching:** 8-week project with 67 students; Summer School with 30 students; theses	**Teaching:** 11-week project with 140 students; theses	**Teaching:** theses
	Subject matter experts: 10 workshops with industry participants	**Subject matter experts:** discussion of results with project partners (2 milestone meetings); discussion of results with subject matter experts of a vehicle manufacturer (1 milestone meeting).	**Subject matter experts:** 2 workshops with project partners; discussion of results with subject matter experts from a vehicle manufacturer (1 milestone meeting).	**Subject matter experts:** 3 in-depth workshops on the overall method; panel speaker on automotive security systems engineering
			Conclusion	Final result (framework)

Figure 3-2: Relation of my work to the DSR-PM.

Suggestions: I asked myself how automotive security engineering can be integrated into the early phase of system design. A central aspect of system design is the creation of requirements. An initial idea was to integrate the security aspect into requirements engineering (RE). Based on my project experience, RE is used in every larger company. For a better understanding of requirements, they can be supplemented with models [**Pol16**; **Rup14**].

Development: As part of a scientific work [**Jap20**], I created a solution for extending the activities of requirements engineering (Elicitation, Negotiation, Documentation, Validation/Verification) to include security. The creation of different models served as a basis for the derivation of requirements. Here I used parts of the CONSENS method, which I describe in Section 2.2.3. I extended the procedure from [**Jap20**] to ISO/SAE 21434 in subsequent iterations.

In the context of a scientific work [**JKK20**] of mine, an approach was developed which uses a 3D environment with which the representation of security-related damage scenarios is made possible. This aims to improve the communication between the stakeholders in the concept phase and the quality of the requirements. This approach is part of my solution and is described in Section 5.2.

Evaluation: Over a period of 1.5 years, I conducted 11 workshops on MBSE. The main purpose of the workshops was to teach and apply MBSE content in the concept phase. Usually, the work in the concept phase is done by experts from different disciplines. Basically, there were no participants in any of the workshops who had significant security knowledge. I used the workshops to see how security can be integrated into workshops with participants without significant security knowledge. In Section 6.1 I describe the evaluation of Iteration 1 in detail.

3.2.2 Iteration 2

Awareness of Problem: Through the workshops in Iteration 1, I learned how to apply MBSE in the concept phase. I also got a feeling for which topics can be dealt with to what extent in these workshops and how complex the model constructs should be.

Generally, it was possible to identify valuable components or data (assets) in the workshops. However, the identification of assets is not sufficient for a security risk analysis. Although the participants discussed possible countermeasures, it was not possible to secure the identified assets in any of the workshops.

Suggestions: To protect the identified assets, a risk analysis must be carried out to determine which assets are vulnerable. This helps on the one hand to identify vulnerabilities and on the other hand to distinguish critical vulnerabilities from less critical vulnerabilities and thus to set the right priorities in engineering. To fix critical vulnerabilities, countermeasures are needed. I asked myself how MBSE can support this in the concept phase.

Development: In the context of two scientific works [**JAD21**; **JA21**] I have elaborated the following solutions. In Section 5.5 I present an approach for performing a model-based risk analysis in the concept phase. Based on this, the approach from Section 5.6 is used, which presents the application of security design patterns in the context of the concept phase. Both approaches use simple model constructs, which ensures that an interdisciplinary team of experts without in-depth MBSE knowledge can understand and apply the approaches.

Evaluation: During an 8-week project in the context of teaching with 67 master's students, I evaluated the partial solutions that I had worked out up to that point. Beforehand, I tested these partial solutions in a one-day workshop with 30 master's students. During two milestone meetings with a German premium vehicle manufacturer, I was allowed to present and discuss these results.

The project consisted of the following activities: (1) Using a 3D environment to identify security-critical damage scenarios. The students were able to identify 120 security-critical damage scenarios. In this case, the approach described in Section 5.2 was used. (2) Deriving models from the scenarios and performing a risk analysis. The models had a median of 17 components and 21 relationships. The approach described in Section 5.5 was used here. (3) Deriving requirements from the models. Here I found that the creation of models resulted in a median change in requirements of about 60%. (4) Application of countermeasures using security design patterns. In the median, approx. 24% of the components and relationships had to be adapted for this. The approach described in Section 5.6 was used here.

In the Sections 6.2 and 6.3 I describe the evaluation of Iteration 2 in detail.

3.2.3 Iteration 3

Awareness of Problem: Based on the evaluation of Iteration 2, I was able to identify the following challenges. The requirements engineering activities are not sufficient as a basis for integrating the security aspect in automotive companies during the concept phase. In addition, I was able to determine that the selection of countermeasures provided was not sufficient in the aforementioned project.

Suggestions: At the beginning of 2021, ISO/SAE 21434 was available as a Draft International Standard (ISO/SAE DIS 21434). Based on literature research, expert blogs and discussions with subject matter experts, ISO/SAE 21434 was a promising successor to SAE J3061 [**SAE16**]. Furthermore, similar to ISO 26262[**ISO18**] in the safety area, ISO/SAE 21434 should become the new standard in the security area. For the concept phase, ISO/SAE (DIS) 21434 describes, with the help of numerous complex inter-related requirements, which work products must be created so that a cybersecurity concept can be created.

Development: ISO/SAE 21434 does not define a concrete normative procedure. To comply with ISO/SAE 21434, the automotive companies must develop their own procedure. To

provide a concrete procedure for the concept phase, I used the partial solutions I had worked out up to that point. In doing so, I have made adjustments and extensions to my partial solutions so that they are compliant with the ISO/SAE DIS 21434 available at that time. In Iteration 4, I describe my final solution, which conforms to the final ISO/SAE 21434. Apart from ensuring conformity to ISO/SAE DIS 21434, further security design patterns were developed as part of a master's thesis I supervised [**Fah21**], which are mentioned in Section 5.7.

Evaluation: Within the context of an 11-week teaching project with 140 master's students, I evaluated the partial solutions that had been developed up to that point. The initial preparation was a master's thesis [**Fah21**] supervised by me, in which, among other things, a continuous application example was developed, the development of which conformed to ISO/SAE DIS 21434. In the context of a milestone meeting with a German premium vehicle manufacturer, I was allowed to present and discuss the results developed in the context of my work.

The 11-week project consisted of the following activities: (Activity 1) Use Cases and associated Damage Scenarios and Threat Scenarios were identified. (Activity 2) The relevant components and relationships of a system architecture required to realise the scenarios identified in Activity 1 were modelled. (Activity 3) In this activity, a model-based risk analysis was performed. Based on this, decisions were made on how to deal with the risks. (Activity 4) If the risks were high, a countermeasure should be chosen to minimise the risk. (Activity 5) To check the effectiveness of the countermeasure, a new risk analysis of the changed system components and their component relationships was carried out. (Activity 6) Lastly, requirements were derived from the model-based risk analysis.

Activities 1-3 were based on the approaches described in Sections 5.5. Activity 4 was based on the approaches based in Section 5.6 and 5.7. Activity 5 was new. Here the focus was on performing the risk analysis for elements affected by the application of the countermeasure. Activity 6 was based on the approach described in Section 5.8.

In Section 6.4 I describe the evaluation of Iteration 3 in detail.

3.2.4 Iteration 4

Awareness of Problem: The evaluation in Iteration 3 brought several points to my attention, which I will describe below.

Basically, the application of countermeasures, especially in the form of security design patterns, causes a change in the item definition. This includes changing the components and relationships in the item. Furthermore, additional environment elements and environment systems interacting with the item can be added. This requires a new risk analysis including the creation or adaptation of all 15 work products of the concept phase.

In addition to my existing partial solutions, I have found that the participants in the workshops had difficulties in comparing the relative frequency of different damage scenarios.

In the context of the student projects, the partial solutions of my work have so far been implemented mainly with office software. One advantage of this approach was the broad applicability of the partial solutions, as practically everyone had access to such software. A major disadvantage is that this software does not support traceability between models and between models and requirements. This makes change tracking more difficult and has the potential for errors, since the same objects or texts have to be copied and changes to the objects or texts have to be made manually at all points for consistency.

Since students are usually not yet subject matter experts, there is potential doubt about the trustworthiness of the identified vulnerabilities and the correct application of my partial solutions.

Suggestions: The application of countermeasures is required in the detailed system design phase of ISO/SAE 21434. In contrast, the concept phase of ISO/SAE 21434 only requires the selection of countermeasures, but not their application. Since the focus of my work is on the concept phase, I limit the final overall solution of my work to the selection of countermeasures. The implications of the application of countermeasures would, in my view, go beyond the scope of this work.

To facilitate the impact rating for damage scenarios, accident data from the Federal Statistical Office should be used.

To enable traceability, my approach should be implemented using a continuous and realistic application example in a professional MBSE tool.

To check and increase the trustworthiness, I would like to present my approach in several workshops that build on each other and discuss possible improvements in these workshops with experts.

Development: In Section 5.3, I present an approach for the data-driven assessment of damage scenarios. For this purpose, I use data from the Federal Statistical Office, with more than 10 million registered traffic accidents. For this, I aggregated the data and applied it to the ASIL risk classification scheme (cf. Section 5.3).

Based on my project experience, diagrams in the concept phase are often initially created in Office tools. With increasing complexity of the diagrams, change tracking is no longer maintainable without errors. In Section 5.4 I describe in generalised form the result of a development project I led with a German premium vehicle manufacturer. The creation of the initial item definition was implemented at the vehicle manufacturer with the help of the Office tool Visio. Diagrams in the simple modelling language ECML were used, which I describe in Section 2.2.4. In order for the item definition to be used and extended in different parts of the company, a manual conversion to SysML (cf. Section 2-10) was necessary. To reduce the engineering effort and for traceability purposes, a tool was

developed that extracts ECML diagrams from Visio files and generates SysML models from them in a professional MBSE tool.

In Section 5.9 I describe the general procedure of my final approach to the creation of the 15 work products of the concept phase according to ISO/SAE 21434. In particular, I describe the points in my approach where I integrate the partial solutions from Iterations 1-4 mentioned so far. In Section 6.6.1 I explain my final approach by means of a concrete and continuous application example. In particular, I show the implementation of the approach in a professional MBSE tool. This enabled me to ensure traceability between the 15 work products of the concept phase. To simplify the creation of the work products, I have created a SysML profile conforming to ISO/SAE 21434 (cf. Section 6.6.8).

Evaluation: I evaluated the approach for the data-driven assessment of damage scenarios in two workshops with experts from the fields of product development and security engineering (cf. Section 6.5.2).

The model transformation tool was evaluated in a project lasting several months with operational data from a vehicle manufacturer (cf. Section 6.5.3).

With the help of a real life test, I wanted to make sure that the application example is realistic in order to better understand the problems in modelling and specification and to improve the approach on this basis (cf. Section 6.6.9). For this, I tried to think from an attacker's point of view and find suitable tools for carrying out an attack.

I evaluated my final approach in three workshops with security experts over a period of one year (cf. Section 6.6.10). I used the application example from the practical test as a basis. The focus of the first workshop was on dealing with the steps from the identification of use cases to the evaluation of damage scenarios. The second workshop dealt with the steps leading to the modelling and assessment of possible attacks. In particular, the derivation of cybersecurity goals and the selection of cybersecurity controls were covered. The third workshop mainly served to implement the feedback from the previous workshops and for the final evaluation.

In addition to the evaluation of my overall approach, I had the opportunity to share my expertise in automotive security systems engineering as a panel speaker with 60 experts from the automotive sector.

Conclusion: The outcome of my work is a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*.

4 State of the art

In Section 4.1 I mention existing works that have a high relevance to the requirements presented in the Section 2.5. I address the degree to which these works satisfy or do not satisfy the requirements stated. In Section 4.2 I give an overview of the evaluation of the approaches examined. From this I derive the necessity of my work.

4.1 Considered approaches

4.1.1 ThreatSurf: Threat surface assessment

In [**ZPR+22**] an approach for attack surface assessment is presented. This approach includes the steps of asset identification, threat scenario identification, attack path analysis and attack feasibility assessment according to ISO/SAE 21434. Additionally, an automation of these steps is proposed in the paper.

The paper contains the following contents: (1) A general reference architecture that can be mapped to a variety of modern E/E architectures. (2) A comprehensive set of assets in modern vehicles that form the attack surface. (3) Attack steps with associated feasibility analysis in accordance with the requirements of ISO/SAE 21434. (4) An algorithm for automatically generating and evaluating attack paths using the attack steps and attack feasibility. (5) An example application of automated attack surface assessment to various threats from UN Regulation 155.

Assessment: (R1) The approach supports the creation of the majority of the concept phase work products. In particular, no impact analysis is carried out for damage scenarios. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) It is a table-based approach that does not use models. (R4) In particular, the approach has not been implemented in an MBSE tool. (R5) The approach has been briefly described using three realistic threat scenarios (Manipulate Torque, Immobilise Car, Flash Compromised Firmware). No continuous application of the approach is shown on these examples.

4.1.2 Attack surface assessment

In [**PZG+21**] an approach for attack surface assessment is presented. Threat and risk analysis is an important part of the engineering process for automotive cyber security. A key aspect is the identification of the attack surface with a comprehensive feasibility assessment of possible attacks for each asset of a modern vehicle. In this paper, numerous attack steps from the automotive domain are presented and evaluated according to the Attack-Potential Based Approach. With the help of the attack steps, complex attack paths can be modelled. The evaluation of the attack steps that has already been carried out reduces the engineering effort.

Assessment: (R1) The approach mentions damage scenarios, however, it does not elaborate on them. In particular, no impact analysis is carried out for damage scenarios. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) It is a table-based approach that does not use models. (R4) In particular, the approach has not been implemented in an MBSE tool. (R5) The approach has been consistently described using three realistic threat scenarios (Electric Driving, Conductive Charging, Over-the-Air Firmware Update).

4.1.3 TARA+ for L3 automated driving systems

In [**BAS+19**], a model is proposed for the analysis of cyber security of level 3 systems for automated driving by integrating aspects of functional safety. The model takes into account the probability and impact of an attack and combines them to derive a risk value. The novelty lies in the integration of a formula to calculate the probability of an attack.

Assessment: (R1) The approach uses the attack potential approach to assess attacks and supports the assessment of damage scenarios. In particular, the approach supports the calculation of the risk for different attacks. The approach requires several steps prior to the concept phase, which are not addressed in the approach. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) It is a table-based approach that does not use models. (R4) In particular, the approach has not been implemented in an MBSE tool. (R5) The approach has been described throughout using a realistic example (Highway Chauffeur).

4.1.4 SARA: Security automotive risk analysis method

In [**MBZ+18**] the SARA framework is presented, which consists of four parts. (1) The feature definition describes the defence scope of the assessed system. The system definition is based on two architectures. The physical architecture represents interfaces, controls, sensors, actuators and communication links. The logical architecture represents the data flows. By knowing the data flows, the expert can determine the severity of attacks on the system's assets. (2) The threat specification describes the mapping from a threat to a threat model. (3) The risk assessment contains the risk determination for identified attacks. The probability of attack is identified with the help of various parameters. With the help of attack trees, possible attack paths are modelled and the easiest attack to realise is identified. (4) Countermeasures are used to reduce the calculated attack risk. After applying a countermeasure, a new risk analysis must be carried out. Steps 1-4 are repeated until the risk is reduced to an acceptable level.

Assessment: (R1) The approach supports the creation of most of the work products of the concept phase with the help of a comprehensive process model. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) The approach uses models, but does not use a standardised modelling language from the

MBSE area. (R4) In particular, the approach has not been implemented in an MBSE tool. (R5) The approach has been described throughout using two realistic examples (Vehicle Tracking and Comfortable Emergency Brake Failure).

4.1.5 Attack surface analysis

Connected autonomous vehicles are equipped with autonomous functions supported by new sensor and communication capabilities. Such functions enable new types of attacks. Countermeasures must be developed for this. One possible approach is to use reference architectures to analyse the attack surface. Existing approaches are either too simple for sufficiently detailed modelling or require the specification of too many details. In [**MBL19**], a reference architecture for connected autonomous vehicles is presented. This enables a holistic analysis from a functional point of view. Furthermore, partial reference architectures for communication, e.g. with the cloud, are presented. This enables the analysis from a communication perspective.

Assessment: (R1) The approach does not include a process model, but takes into account many activities of the concept phase. Damage scenarios and countermeasures are not considered. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) The approach uses several detailed and consistent models. For an approach that focuses on a reusable reference architecture for connected autonomous vehicles, the use of a standardised modelling language from the field of MBSE is missing. (R4) In particular, the approach has not been implemented in an MBSE tool. This appears contradictory to the complexity of the numerous detailed and interrelated models. (R5) The approach shows in great detail and with continuous examples the application of reference architectures for attack surface analysis of networked autonomous vehicles.

4.1.6 THREATGET: Automated attack tree analysis

According to [**CCS+23**], the automotive industry is evolving from simple, isolated vehicles to networked, heterogeneous systems that form a complex traffic infrastructure. The additional means of communication lead to larger attack surfaces that can be exploited by both physical and remote attackers if not carefully protected. This exposes the automotive sector to new cyber risk factors. Several research projects have developed approaches to secure vehicles and infrastructures by identifying and mitigating potential threats to the automotive sector. [**CCS+23**] builds on the developments from these projects and related standards and regulations. The core of the work is the threat modelling tool THREATGET. THREATGET can be used to collect and analyse information about threats. Furthermore, it enables risk assessment and the automation of routine tasks.

Assessment: (R1) The approach considers most of the work products of the concept phase, but does not go into detail about them. (R2) The approach does not consider cybersecurity goals and requirements. (R3) The approach uses a Domain Specific Language (DSL) to

model the system architecture. Furthermore, the approach uses data flow diagrams, which are known in the security field. (R4) An own modelling and analysis tool (THREATGET) was developed to realise the approach. (R5) The approach is partly illustrated by means of a realistic example (Remote Attack on the Brake Control).

4.1.7 Multi-concern development lifecycle

According to [**SWS18**], synergies between different development processes for complex, networked and intelligent cyber-physical systems are necessary to ensure product quality and reduce time to market. Often, developing a product according to a standard provides the means to meet a quality attribute. Compliance with standards usually means extra work for product development. Compliance with multiple standards compounds this problem. Since cyber-physical systems are networked and highly automated systems, the combination of safety and cybersecurity is of great interest. In [**SWS18**] synergies between the safety standard ISO 26262 and the cybersecurity standard ISO/SAE 21434 are identified. This avoids duplication of the same or very similar activities by different engineering teams.

Assessment: (R1) The approach considers most of the work products of the concept phase. No damage scenarios and threat scenarios are considered. (R2) The approach shows the derivation of cybersecurity goals and requirements. (R3) It is a table-oriented approach without using a modelling language. (R4) The approach was not implemented with an MBSE tool. (R5) The approach was represented by a realistic but not detailed example (Satellite Navigator Receiver).

4.1.8 Model-based attack tree generation

Networked and highly automated driver assistance systems require interfaces to the outside world and within the vehicle. These interfaces can potentially be used for cybersecurity attacks. In this context, it is necessary to investigate possible attack paths. The creation of these attack paths is labour-intensive. In [**KLB+21**], an approach for semi-automatic attack tree generation is presented. This reduces the modelling effort. The approach takes into account the attacker motivation and the functional dependencies of the initial system architecture from the concept phase.

Assessment: (R1) The approach takes into account several but not all activities of the concept phase (item definition, asset identification and the modelling of attack paths), which is additionally described by a process model. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) The approach does not use a standardised modelling language. The models presented are similar to SysML, especially the SysML Internal Block Diagrams. The models additionally contain pictograms and a color scheme. (R4) The approach was prototypically implemented using the Eclipse Modeling Framework (EMF). EMF is an established MBSE platform in the

open source area. (R5) The approach is illustrated throughout using a realistic example (Road Speed Limiter).

4.1.9 Mutually supporting safety and security analyses

Failures in cyber-physical systems, such as cars, are caused by faults or attacks. Faults are dealt with in the area of safety engineering. Attacks are dealt with in the field of security engineering. Both disciplines use their own terminology, procedures and tools. However, both disciplines use a common system architecture and use e.g. fault trees and attack trees for the analyses. In [**KMA21**], the coordination of content between both disciplines is dealt with on the basis of models. An existing software tool (YAKINDU Security Analyst) was further developed for this purpose. The software tool allows both disciplines to benefit from each other's analyses, which improves the consistency, comprehensiveness and alignment of the disciplines.

Assessment: (R1) Using the software tool YAKINDU Security Analyst, most of the work products of the concept phase are supported. (R2) The approach does not show the derivation of cybersecurity goals or cybersecurity requirements. (R3) The approach does not use a standardised modelling language. One type of model is similar to the SysML Internal Block Diagrams. According to the approach, a separate Domain Specific Language is used. (R4) The approach was implemented in the software tool YAKINDU Security Analyst, which is partly used in the automotive industry. (R5) The approach is illustrated throughout with a realistic example (Automated Steering with Software Update via USB).

4.1.10 Model-based safety assessment with SysML

According to [**MN20**], mastering the complexity of modern software-intensive systems, e.g. in the field of automotive engineering, is a challenge from a safety engineering perspective. Model-based safety analysis techniques show promising results for overcoming this challenge by automating the generation of the required artefacts for a safety proof. In [**MN20**] an approach is presented that extends SysML models with Component Failure Trees (CTF) to support Failure Mode and Effects Analysis (FMEA). While most existing approaches based on CFTs only target the system topology, the approach describes an integration of CFTs with SysML Internal Block Diagrams as well as SysML Activity Diagrams and realises it in a software tool.

Assessment: (R1) The approach comes from the field of safety engineering. As in security engineering, an item definition is created. Instead of attack trees, CTFs are created and analysed. (R2) The approach does not show the derivation of goals or requirements. (R3) The approach uses SysML as the standardised MBSE modelling language. (R4) The approach has been implemented on the JetBrains Meta Programming System platform. (R5) The approach is illustrated throughout using a realistic example (Electronic Power

Steering/Boost Recuperation System).

4.1.11 Security-driven automotive development lifecycle

ISO/SAE 21434 provides a generic framework for considering automotive cyber security across the entire life cycle. In [**DME+21**] an approach is proposed to create the work products required by ISO/SAE 21434. The proposed life cycle model complements ISO/SAE 21434 and forms the basis for the company-specific specifications. A fundamental feature of the approach is the central role of threat modelling, vulnerability assessments and the derivation of cyber security requirements at both system and subsystem levels. The approach proposes design guidelines that are sufficiently concrete for engineers, yet generic enough for company-specific adaptations and refinements. The approach has been designed to be compatible with Automotive SPICE, an established process framework in the automotive industry.

Assessment: (R1) The approach has a detailed process model for creating the work products of the concept phase. (R2) The approach mentions the derivation of cybersecurity goals and requirements in the process model, but does not address them further. (R3) The approach uses a type of diagram that is unknown in the MBSE area but known in the security area (Data Flow Diamgrams). (R4) The approach was not implemented with any MBSE tool. One diagram was apparently implemented with Microsoft Threat Modeler, a tool for modelling Data Flow Diagrams. (R5) The approach was illustrated with a realistic example (Road Wheel Steering Control Unit). The example only addresses some of the steps mentioned in the process model.

4.1.12 HEAVENS 2.0: An automotive risk assessment model

HEAVENS is a known approach in the field of research and industry for risk assessment. However, this approach does not fulfil all the requirements for risk analysis defined in ISO/SAE 21434. HEAVENS 2.0 is presented in [**LAO21**]. This approach contains 17 improvements to the original approach and is compliant with ISO/SAE 21434. The basis for the improvements was an analysis of the gap between HEAVENS and the risk analysis according to ISO/SAE 21434. Furthermore, known weaknesses of HEAVENS served as a basis for improvement. HEAVENS 2.0 makes it easier for users of HEAVENS in particular to apply ISO/SAE 21434.

Assessment: (R1) The approach has a detailed process model for creating the work products of the concept phase. (R2) The approach mentions the derivation of goals in the process model. The consideration of requirements is not part of the approach. (R3) The approach does not use a modelling language standardised in the MBSE area. However, a type of diagram known in the security field (Data Flow Diamgrams) is used. Furthermore, attack trees are used to model multiple attack paths. (R4) The approach was not implemented with any MBSE tool. (R5) The approach was illustrated consistently

with a realistic example (Speed Limiter).

4.1.13 Cybersecurity threat analysis

According to [**DES+21**], the integration of cybersecurity into the development processes of the automotive industry is not yet mature. ISO/SAE 21434, which forms the consensus regarding cybersecurity in the automotive industry, can serve as an aid. ISO/SAE 21434 describes the requirements for the processes. The realisation of the processes must be carried out by the companies.

In [**DES+21**], concrete steps and methods for the realisation of ISO/SAE 21434 are proposed to help engineers to integrate secure system design techniques and systematic approaches for the elicitation of cybersecurity requirements into their development processes. The aim of the approach is to develop a generic security-oriented development cycle model.

Assessment: (R1) The approach goes into detail about the creation of the work products of the concept phase. (R2) The approach shows the derivation of goals and requirements. (R3) No standardised modelling language from the MBSE area is used to model the system architecture. For modelling the attack paths, an own variant of fault trees is used. Furthermore, Data Flow Diagrams, which are known in the security field, are used. (R4) The approach was mainly implemented in a spreadsheet program. (R5) The approach was illustrated in detail throughout with the help of a realistic example (Steering Control Unit). Some steps are indicated in an overview figure, but were not readable.

4.1.14 Automotive SPICE for cybersecurityassessment model

In [**MEM+22**] an approach is described which extends Automotive SPICE, a process framework, with regard to cybersecurity. For this purpose, the process groups Security Requirements Management, Security Risk Management, Security Design and Implementation and Security Testing and Verification were extended to the existing Automotive SPICE process framework. ISO/SAE 21434 served as the basis for the extension of the process framework.

The authors state that the approach enabled a comprehensive and systematic assessment of the vehicle's cybersecurity and that the use of Automotive SPICE models and tools enabled them to perform the assessment effectively and to document the results clearly.

Assessment: (R1) The approach goes into detail about the creation of the work products of the concept phase. (R2) The approach shows the derivation of cybersecurity goals and requirements. (R3) No standard MBSE modelling language is used to model the system architecture. (R4) The approach was mainly implemented in a spreadsheet program. (R5) The approach was illustrated consistently in detail with the help of a realistic example (Key Less Go System).

R1	Support in the creation of the work packages of the concept phase of ISO/SAE 21434
R2	Systematic approach for deriving requirements
R3	Use of a standardised modelling language from the field of systems engineering
R4	Realisation in a professional MBSE tool or based on a professional MBSE platform
R5	Realistic and continuous example from the automotive domain

To what extent do the appraoches considered satisfy the requirements? Satisfied / Partially s. / Not satisfied		**Requirements**				
Considered approaches		**R1**	**R2**	**R3**	**R4**	**R5**
[ZPR+22]	ThreatSurf: Threat surface assessment					
[PZG+21]	Attack surface assessment					
[BAS+19]	TARA+ for L3 automated driving systems					
[MBZ+18]	SARA: Security automotive risk analysis method					
[MBL19]	Attack surface analysis					
[CCS+23]	THREATGET: Automated attack tree analysis					
[SWS18]	Multi-concern development lifecycle					
[KLB+21]	Model-based attack tree generation					
[KMA21]	Mutually supporting safety and security analyses					
[MN20]	Model-based safety assessment with SysML					
[DME+21]	Security-driven automotive development lifecycle					
[LAO21]	HEAVENS 2.0: An automotive risk assessment model					
[DES+21]	Cybersecurity threat analysis					
[MEM+22]	Automotive SPICE for cybersecurityassessment model					

Figure 4-1: Evaluation of the examined state of the art against the thesis requirements.

4.2 Literature rating

A comparison of the state of the art with the derived requirements from Section 2.5 results in the following rating, which is summarised in Figure 4-1:

R1) ISO/SAE 21434 concept phase work products: The following approaches support the creation of most of the concept phase work products: The [**DME+21**] approach assists in creating the preliminary system architecture, identifying assets, identifying security threats, performing a risk assessment and selecting mitigations. Although the approach mentions the derivation of cybersecurity goals and requirements, this is not covered further. The approach [**LAO21**] supports the creation of the preliminary system architecture, the identification of security threats and the risk determination. The approach supports the derivation of cybersecurity goals, but not the derivation of requirements. The approach [**DES+21**] supports the identification of assets, the description of damage scenarios, the identification of threat scenarios, the determination of attack paths, the risk determination and the selection of mitigations. The approach also supports the derivation of cybersecurity goals and requirements. The approach [**MEM+22**] supports the creation of the preliminary system architecture, the identification of security threats, the identification of damage scenarios, the risk determination and the selection of mitigations. The approach also supports the derivation of cybersecurity goals and requirements. Unfortunately, none of these approaches uses a standardised modelling language from the MBSE area. Likewise, none of these approaches has been implemented or applied in an (MBSE) modelling tool.

R2) Cybersecurity goals and cybersecurity requirements: The following approaches support the derivation of cybersecurity goals. Some of these approaches also support

the derivation of cybersecurity requirements. The approach [**SWS18**] is table-based. The derivation of cybersecurity goals is based on the Attack Potential Based Approach. The approach [**DME+21**] describes the context in which cybersecurity goals have to be created, but this is not shown in the application example. The approach [**LAO21**] addresses most of the work products of the concept phase and especially considers the derivation of cybersecurity goals, but not the derivation of cybersecurity requirements. The approaches [**DES+21**; **MEM+22**] support the creation of most work products of the concept phase. In particular, these approaches support the derivation of cybersecurity goals and cybersecurity requirements. Unfortunately, none of the approaches uses a standardised modelling language from the MBSE domain. In addition, none of the approaches has been implemented in an MBSE tool.

R3) MBSE standardised modelling language: Those approaches that use a modelling language use a Domain Specific Language (DSL), which can only be understood by specific experts. In addition, the use of a DSL makes it difficult to compare the results between different departments and companies if a different modelling language is used in other departments or companies. Only [**MN20**] uses a standardised modelling language from the MBSE area. Although this approach fulfils some of the requirements, it does not address the security domain.

R4) Realisation in an MBSE modelling tool: The [**CCS+23**] approach uses a tool tailored to the security sector. The tool was developed as part of research and is now available commercially. With the help of [**KLB+21**], attack trees can be generated. This is a development that uses the Eclipse Modeling Framework (EMF). EMF is an open source platform for the creation of modelling tools. The approach [**KMA21**] is an extension of the commercial tool Yakindu Security Analyst (YSA). YSA is a tool tailored for the security domain. The approach [**MN20**] is a development based on the JetBrain Meta Programming System (JetBrain MPS). JetBrain MPS is a platform for the development of Domain Specific Languages. The approach uses SysML as a standardised modelling language from the MBSE domain. None of the approaches support the derivation of cybersecurity goals or requirements.

R5) Application by means of a consistent and realistic example: All approaches were illustrated with one or more realistic use cases from the automotive sector. The majority of these approaches used a consistent example. Most of these approaches address only a few work products from the concept phase. Only the following approaches used an MBSE tool. In [**CCS+23**], a system architecture is created for the use case Remote Attack on the Brake Control, assets are identified, a risk analysis is performed and an attack tree is derived. In [**KLB+21**], a system architecture is created for the use case Road Speed Limiter, assets are identified and an attack tree is derived. In [**MN20**] the use cases Electronic Power Steering and Boost Reuperations System are addressed. This approach fulfils several of the stated requirements. The security-relevant requirements were not addressed. In [**KMA21**], the use cases Automated Steering and Software Update via USB were addressed. The Yakindu Security Analyst (YSA) tool is used for this. With the help of YSA, most of the

work products of the concept phase can be created. Unfortunately, YSA is text-based and only offers the generation, but not the editing of models. This makes YSA in particular unsuitable for use in the concept phase by an interdisciplinary team of subject matter experts.

None of the approaches examined, nor any combination of existing approaches, fully meets all requirements. A crucial disadvantage is the lack of support for the derivation of cybersecurity goals and requirements from models. Only one approach used a standardised modelling language from the MBSE area. The other approaches were either not model-based or used a Domain Specific Language, which could only be understood by security or modelling experts, but not by an interdisciplinary team of subject matter experts as part of the concept phase. Although many approaches used models, only four were implemented in an MBSE tool. Most approaches were implemented and demonstrated using a realistic example. Unfortunately, most examples were only partially described using models. Only a few approaches were able to ensure digital continuity between the work products of the concept phase. Overall, there is a need for action to provide a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering*.

5 Developing a Cybersecurity Concept According to ISO/SAE 21434

The need for action that emerges from Chapter 4 shows that there is currently no *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering* that satisfies the requirements identified in Chapter 2. In this work, I present such a framework. For this purpose, I give an overview of the framework in the Section 5.1. Subsequently, I present the parts of the framework in the Sections 5.2 - 5.8. Finally, in the Section 5.9, a process model is presented that connects each of the interrelated parts. Then, in the Section 6, the evaluation of the framework is presented.

5.1 Overview of the framework

My work consists of several parts and a process model that connects the individual parts with each other (cf. Figure 5-1). The result of my work is a process model for the development of a cybersecurity concept according to ISO/SAE 21434.

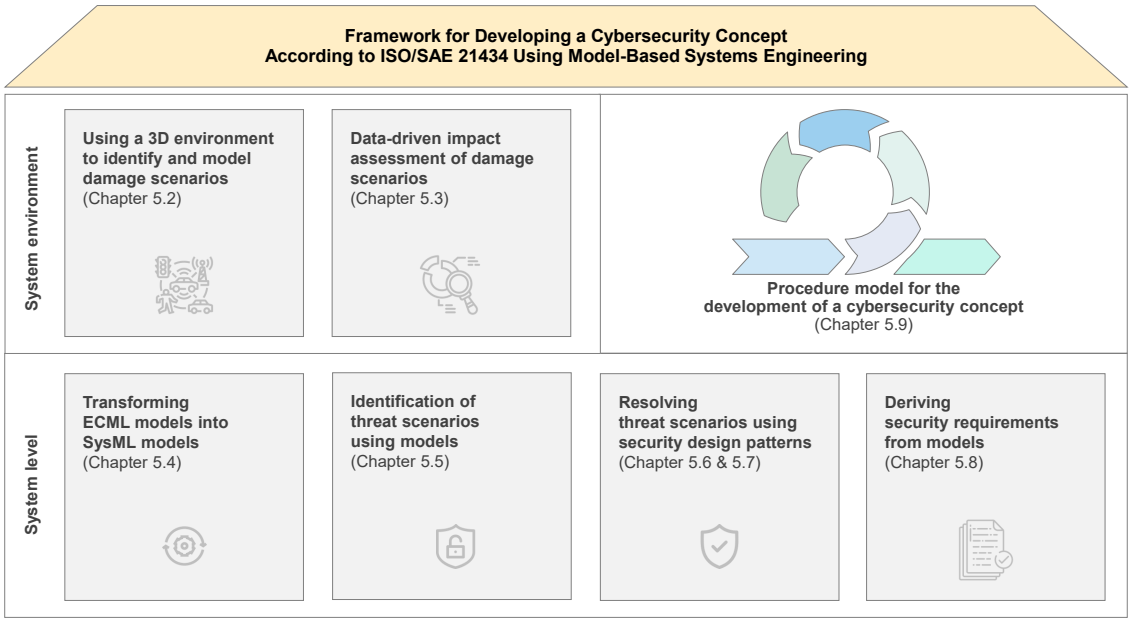


Figure 5-1: Components of the framework.

According to ISO/SAE 21434, damage scenarios must be identified as part of the concept phase. A damage scenario is an adverse consequence involving a vehicle or vehicle function and affecting a road user. In Section 5.2 I present an approach that supports the identification and modelling of damage scenarios using a 3D environment. With the help of the 3D environment, potential hazards can be identified and damage scenarios can be run through and thus better understood.

According to ISO/SAE 21434, the damage scenarios must be assessed according to different categories with regard to the possible negative effects on road users. In Section

5.3 I present an approach that supports the assessment of damage scenarios regarding the safety aspect using statistical accident data. The use of statistical data improves the objective assessment of damage scenarios and reduces misjudgements.

According to ISO/SAE 21434, an idem definition must be created. An item is a component or set of components that implements a function at the vehicle level. The Effect Chain Modelling Language (ECML) is a simple modelling language for creating the item in the concept phase. In order to use ECML models in detailed system design, these models must be converted into SysML models. In Section 5.4 I present an approach that enables the model transformation of ECML models into SysML models. This eliminates the effort of manual model transformation. In addition, errors that can occur during manual model transformation are avoided.

ISO/SAE 21434 requires the identification of threat scenarios. A threat scenario is a possible cause for compromising the cybersecurity properties of one or more assets in order to realise a damage scenario. In Section 5.5 I present an approach to identify threat scenarios based on SysML models.

Cybersecurity controls are risk mitigation measures. ISO/SAE 21434 requires that cybersecurity controls for critical threat scenarios are selected in the concept phase. Security Design Patterns (SDPs) are solution patterns for recurring security design problems and are used for risk mitigation. In Section 5.6 I present an approach that enables the use of SDPs based on SysML models. In Section 5.7 I relate to an elaborated initial catalogue of SDPs.

ISO/SAE 21434 requires that security requirements are created in the concept phase. In Section 5.8 I present an approach that explains the derivation of security requirements based on SysML models.

In Section 5.9 I present a process model for the creation of the cybersecurity concept. The process model consists of 5 phases. I explain which work products are created in which phase. Furthermore, for each phase, I explain which of the described parts are used in which phase for the creation of individual work products. Depending on the focus, the work products of the individual phases are developed by different groups of subject matter experts who communicate with each other. In the context of this work, these are the groups Product Development (Phase 1 and 4), Safety Engineering (Phase 2) and Security Engineering (Phase 3 and 5).

5.2 3D environment for identification of damage scenarios

In this section I present the content of two consecutive scientific papers [**JKK20**; **JSK+22**]. The two papers have three contributions: A 3D environment was developed to identify use cases and damage scenarios at the vehicle level (C_1). To check the feasibility, I developed the first prototype of 3D Engineer (3DE). 3DE provides three-dimensional visual support and can be used by untrained participants in online workshops. Furthermore, the tool

generates a SysML model in the background without requiring SysML knowledge from the participants. The model generation reduces the manual conversion effort and the workshop results can be used directly in professional modelling tools. This prototype was further developed in several student theses supervised by me [**Sch20**; **Kor20**; **3DE21**; **Sch21a**; **Sch21b**; **Var22**]. Further development consisted of the development of new experimental features. In further work, most of the features were aggregated. In addition, the stability and usability of the prototype were improved. Two student assistants were responsible for this development work, which I led. Furthermore, I developed a method that allows a step-by-step identification of use cases and damage scenarios in the 3D environment (C_2). This method enables the transition between the modelled use cases and damage scenarios in the 3D environment to model-based systems engineering with the help of the prototype. I evaluated the approach in 14 online workshops with a total of 118 participants over a period of 3 years (C_3). The main driver for the online orientation was the COVID-19 pandemic [**WHO23-ol**].

5.2.1 Conducting workshops as part of the concept phase

The development of intelligent technical systems such as autonomous vehicles is characterised by close collaboration between different disciplines such as mechanical engineering, electrical engineering and software engineering [**GRS14**]. Collaboration may also extend across different companies, suppliers and subsidiaries, as is common in the automotive sector [**JA21**]. At the beginning of product development, different use cases for the system to be developed are identified during the concept phase. The identification of such use cases is done collaboratively in workshops with different stakeholders. The goal is to create a common understanding. The overall understanding is achieved through the identification of use cases, the delimitation of the system and the creation of a general system architecture [**ISO21**]. Stakeholders contribute their expertise to the common understanding between all workshop participants. It is important to maintain the appropriate level of abstraction in the concept phase; discipline-specific details make common understanding more difficult [**JA21**]. In the context of concept-level workshops, top executives are usually involved, or additional subject matter experts are invited for specific topics [**Jap20**]. Different drivers shape collaboration in workshops. Increasing digitalisation enables stakeholders from different locations and time zones to work together in online workshops using collaborative tools. At the same time, due to the current pandemic, the German government has defined the home office as the first choice of work location in 2021 [**COR21-ol**], so online workshops are often the only choice for collaboration. In addition to use cases, damage scenarios can be identified and addressed at the concept phase [**JA21**]. ISO/SAE 21434 [**ISO21**] defines them as follows: A damage scenario is an adverse consequence affecting a vehicle or a vehicle function that affects a road user (e.g. passenger, pedestrian or vehicle owner). ISO/SAE 21434, which is relevant to the automotive sector, even requires damage scenarios to be considered at the concept phase. Choosing the right tools, combined with an appropriate approach, is critical to successful collaboration in online workshops.

Complicated tools make it difficult to get started, while an unclear or overly comprehensive approach hinders collaboration [**JA21**; **JAD21**].

In Section 5.2.2 I present the literature review. The approach of using a 3D environment to identify use and damage scenarios is presented in Section 5.2.3. The summary of the evaluation and an outlook for future work is presented in Section 5.2.4. The detailed evaluation is described in Section 6.5.1.

5.2.2 Analysis of tools that can be used in the concept phase

In the context of this work, different collaboration tools for use in the concept phase were analysed according to different categories and requirements (cf. Figure 5-2). The tools were generally divided into 2D-based and 3D-based tools. The 2D tools have the advantage that they are easy to understand and use, so that use cases and damage scenarios can be quickly noted and discussed in an online workshop. The disadvantage is that these tools do not address workshop participants' three-dimensional imagination. Trying out the sequences and interrelationships of the use cases and damage scenarios is not visually supported. This shifts the identification of barriers to realisation to later phases of product development, resulting in higher coordination efforts. In contrast, there are 3D tools that address the imagination of stakeholders and partially allow for trial and error of sequences. The disadvantage of these tools is that they can only be used by trained experts such as simulation experts or CAD developers. This hinders the use of these tools in online workshops with interdisciplinary stakeholders. 3D tools usually produce results that can be directly reused in subsequent steps of product development, while 2D tools usually do not allow reuse of results without manual conversion.

R1	Suitable for cooperation in the concept phase
R2	Enables synchronous collaboration
R3	Enables the representation of structural relationships
R4	Enables the representation of behaviour
R5	Supports situational cognition (here: visualization)
R6	Provides low technical barrier to entry (e.g. as web app)
R7	Reduces effort for further use of the results (e.g. by model generation)

To what extent do the tools considered satisfy the requirements?

Satisfied | Partially s. | Not satisfied

			Considered tools	Requirements						
				R1	R2	R3	R4	R5	R6	R7
2D based	Office tools	[GOG22-ol]	Google Docs Editors	■	■	■			■	
		[MSO22-ol]	Microsoft Office	■	■	■			■	
	Workshop tools	[MIR22-ol]	Miro	■	■	■			■	
		[CON22-ol]	Conceptboard	■	■	■			■	
		[COL22-ol]	Collaboard	■	■	■			■	
		[LUD22-ol]	Lucid-chart	■	■	■			■	
		[DRA22-ol]	draw.io	■	■	■			■	■
		[FIG22-ol]	Figma	■	■	■		■	■	■
3D based	CAD tools	[ONS22-ol]	Onshape		■	■		■	■	■
		[FUS22-ol]	Fusion 360		■	■		■		■
		[THI22-ol]	Thinkercad		■	■		■	■	■
	Modeling tools	[SKE22-ol]	Sketchfab			■	■	■	■	■
		[MOD22-ol]	Modelo			■	■	■	■	■
		[TRI22-ol]	SketchUp			■	■	■	■	■
		[VEC22-ol]	Vectary		■	■		■	■	■
		[CER+16]	Co-3Deator	■	■			■	■	
	Simulaiton	[SVL22-ol]	SVL Simulator			■	■	■		■
		[SIM22-ol]	SIMPHERA			■	■	■		■
		[BHJ17]	Office Work Simulator	■	■	■	■	■		■
		[FGH+15]	Driving Simulator	■	■	■	■	■		

Figure 5-2: Analysis and evaluation of the examined approaches.

In the following, I present a categorisation and evaluation of the tools analysed. I make a basic distinction between 2D and 3D based tools and conclude that there is a need for a tool that combines the advantages of both types. I divide the 2D-based approaches into office tools and workshop tools. Office tools are primarily used to create documents such as slides, text and tables. These tools use 2D shapes such as rectangles, circles, etc. for visual communication that can be linked together. 3D-based approaches are divided into CAD tools, 3D modelling tools and 3D simulation tools. CAD tools allow for accurate geometric modelling. Among other things, they can be used to check concepts geometrically at an early stage and also to create models as a basis for production. 3D modelling tools focus on visualisation. This makes it easier to create models for visualising prototypes. 3D simulation tools can be used to test and visualise complex processes. By posting the Figure 5-2 (without full rating) in a career network (1425 views, 10 comments), I was able to find out that the categorisation mainly covers tools for concept development.

In the following I describe the requirements according to which the different tools were rated. The requirements are based on my intensive experience in using different online collaboration tools with industry and research partners during the pandemic. The requirements generally relate to the level of support for collaboration between different stakeholders in

online workshops. *R1: Online collaboration tools for use in the concept phase must not require in-depth expertise for use, e.g. knowledge of geometric modelling or knowledge of modelling simulation processes.* I justify this by pointing out that the common professional basis of the interdisciplinary stakeholders is low. *R2: Online collaboration tools must allow simultaneous editing of a document, an environment, etc., in order to enable parallel work and thus active collaboration of several participants.* This contrasts with the more one-way collaboration of presenting results in an online meeting. *To illustrate complex structures R3 or complex interactive sequences R4, online collaboration tools must provide appropriate means and functionalities.* The creation of structural relationships in concept development is fundamental. If you need to identify use and damage scenarios that represent behaviour, an associated tool must also allow you to model behaviour. *R5: To improve visual relationships, online collaboration tools must provide visualisation. R6: Online collaboration tools must be immediately usable without technical barriers and independent of department and company.* When stakeholders of a company or several companies from different disciplines collaborate online, there are different IT infrastructures with different rights concepts for the use of software. To ensure that collaboration does not fail because of unsuitable application software, the application software must be usable from any operating system, any location, any web browser, with minimal requirements for the rights concept. *R7: In order to be able to reuse results outside of online workshops, online collaboration tools must have an export function that allows further processing of the results on a fine-granular basis.* This means that an image or PDF export, which some of the analysed tools offer, is not sufficient.

I will discuss some of the approaches in more detail below. Microsoft Office [**MSO22-ol**] is suitable for creating Office documents. Documents can be edited by multiple users simultaneously without training. Shapes such as rectangles can be used to create structures using connecting lines. It is not possible to create complex interactive sequences. Those without Office can edit documents in the browser via an invitation link, but the display is not the same or error-free as in the native application. Editable export to a non-Office tool is not possible. Onshape [**ONS22-ol**] is a CAD tool. CAD models can be viewed and moved by various stakeholders. However, editing and creating CAD models requires specific expertise. The tool can be used collaboratively. Each CAD model is made up of individual parts, allowing complex structures to be created. As a web tool, access is easy. Once created, CAD models can be reused in other tools. SIMPHERA [**SIM22-ol**] is a 3D simulation tool for testing critical road traffic situations. Simulations can be viewed by different stakeholders, but creating and editing simulations requires specific expertise. The tool cannot be used collaboratively. Complex structures and processes can be modelled. Access to the tool requires lengthy preparation through installation and may require approval of installation rights by system administrators. Simulation results can be reused in other tools. Overall, none of the tools examined, and no trivial combination of tools, meets all the requirements.

5.2.3 Systematic identification of damage scenarios

In this section I present a method in which 3DE has been used in the context (cf. Figure 5-3). The method is the result of numerous workshops (cf. Section 6.5.1). The method extends the CONSENS method [**GRS14**], which is an approach from model-based systems engineering. The CONSENS method supports the workshop moderator in creating different structural and behavioural models. This includes the creation of scenario visualisations. In the method, I show how the developed tool can be used in scenario visualisation. Furthermore, I describe how damage scenarios can be derived from application scenarios and how they can be linked in terms of modelling. The method is based on my experience as a workshop moderator and the feedback of the workshop participants. The workshop moderator uses the method to guide the participants through the steps to achieve a step-by-step extension and refinement of the results. To do this, the workshop moderator uses guiding questions to support the process (cf. Figure 5-4).

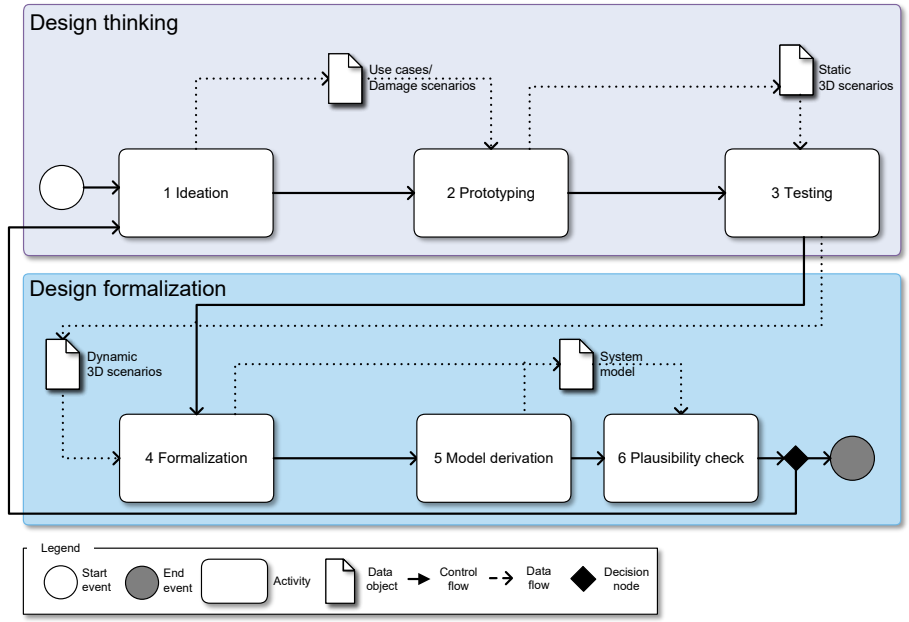


Figure 5-3: General procedure.

1 Ideation	Which use cases and damage scenarios are relevant for the system to be developed?
	How do you rate the priority of each case?
2 Prototyping	Which 3D objects do you need to visualize a case and how do the 3D objects relate to each other?
3 Testing	What important sequences does the selected case contain?
	What impediments do you notice when going through the case?
	What needs to be changed or added so that the sequence can be realized without problems?
4 Formalization	How would you describe the case textually?
	How could you describe the process in more detail using the identified 3D objects?
5 Model derivation	Use the information obtained to derive a system model.
6 Plausibility check	Would you still understand the case description you created and the model relationships you identified in X weeks?

Figure 5-4: Guiding questions for the workshop moderator.

The method consists of two parts. I illustrate the method and the use of 3DE based on the outcome of the online workshop I moderated with 7 product development experts (cf. Figure 5-5 and Section 6.5.1). Part 1 (Design Thinking) serves to determine the expertise of the workshop participants. This requires their active participation in the workshop. Part 2 (Design Formalisation) is used to formalise the workshop results for use in further steps of product development (e.g. requirements engineering/architecture design). A subset of the workshop participants is sufficient for the formalisation. In Part 1 I use process steps from Design Thinking [**GJL+10**]. I focus on the steps Ideate, Prototype and Test. In Step 1 (Ideate), use cases and potential damage scenarios are identified, written down and discussed. For example, the 2D-based workshop tools mentioned in Section 5.2.2 can be used. The discussion will ensure a common understanding between all workshop participants. In order to use the limited time of the workshop participants effectively, the use cases and damage scenarios need to be prioritised so that only relevant cases are worked on. In the workshop with the 7 product development experts, the participants focused on the use case *driving onto road* and derived the damage case *overtaking vehicle not visible*. Step 3 (Prototype) is used to visualise the use cases and damage scenarios. For this purpose the participants use 3DE. First, the 3D objects required for the use case have to be identified and placed in 3DE. By moving the 3D objects, the behaviour of the use cases or damage scenarios can be communicated between the workshop participants in Step 4 (Test). In the workshop the participants modelled the case shown in Figure 5-5. By moving the 3D objects, the participants constructed several sequences and finally identified a concrete damage scenario.

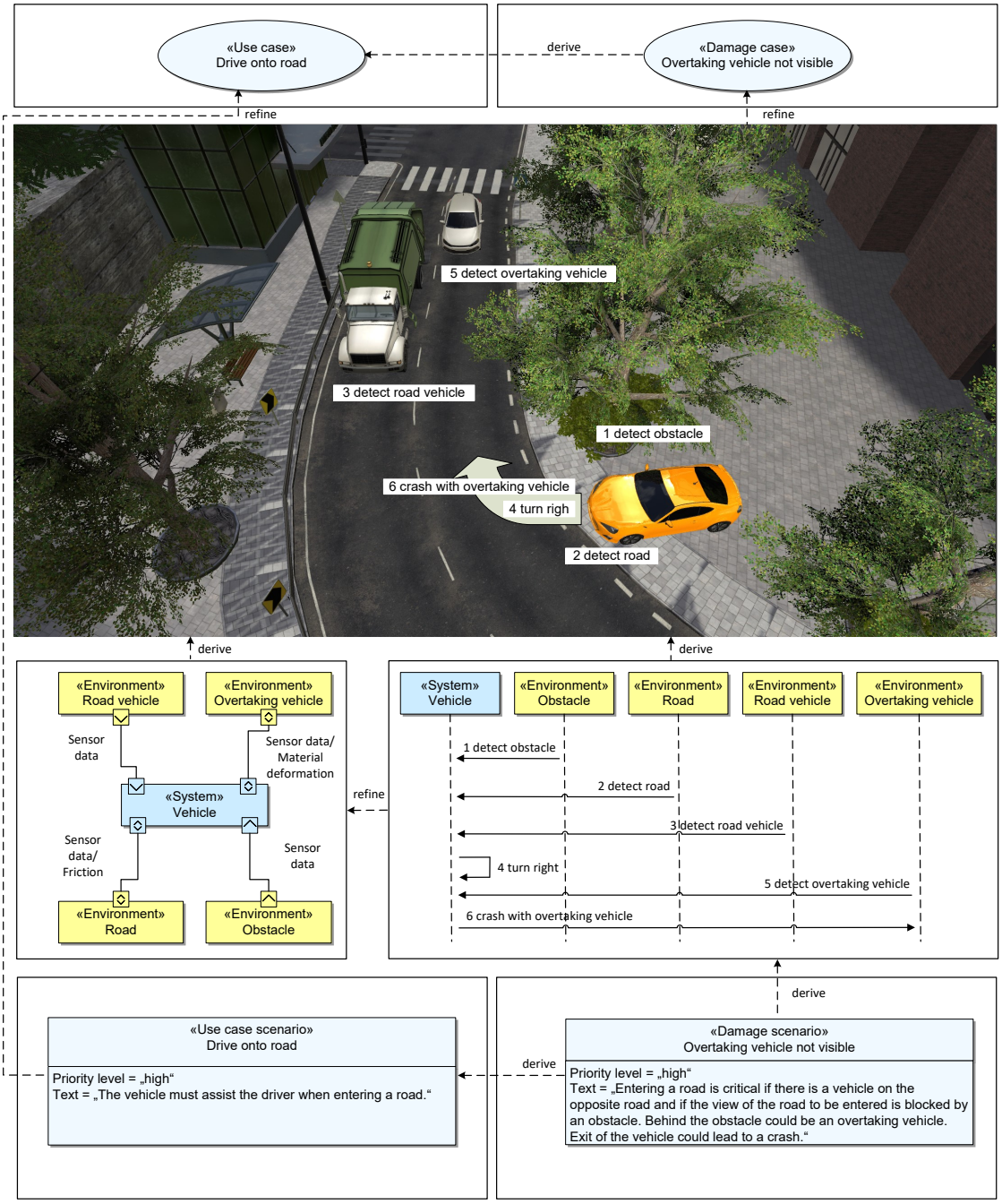


Figure 5-5: Result of the online workshop with 7 experts from the field of product development.

Part 2 involves formalising the use case to ensure the usability of the results outside the workshop. This step can be done with a subset of the workshop participants. In Step 4 (Formalise) the discussed case is formalised based on the experiences in the prototype and testing steps. This is done by describing the object relationships and sequence steps in 3DE. Furthermore, the explicit formulation of the use case and the corresponding damage case is done in some short sentences. In the example (damage scenario) *Entering a road is critical if there is a vehicle on the opposite road and if the view of the road to be entered is blocked*

by an obstacle. Behind the obstacle could be an overtaking vehicle. Exiting the vehicle could lead to a crash. In Step 5 (Derive model) the tool assists the user by automatically deriving SysML models. I chose SysML as the modelling language [**SYS15**] because it is one of the de facto modelling languages in systems engineering [**DOR16**]. The tool generates a black box structural model from the object relationships in the form of a SysML IBD. Based on the sequence information, the tool generates a black-box behaviour model in the form of a SysML sequence diagram. The model generation was first introduced in [**JKK20**]. Step 6 (plausibility check) is a final quality check to ensure the usability of the results outside the workshop. During Step 6 the results can be reviewed by other people or a report can be presented to other people.

5.2.4 Evaluation summary and identified limitations

In 14 online workshops, suggestions were made for improving the approach to identifying use and damage scenarios and for improving 3DE. The first 12 workshops were conducted in the context of teaching and were used for initial evaluation. For further evaluation, two workshops were held with subject matter experts. The following is an extract of the feedback from the subject matter experts and the derivation of future work. In general, the approach was well received by the 17 experts. I would like to highlight the following statement "The approach can be used in industry. Generated SysML models can be refined and the visualisation can be inserted as a screenshot into any requirements engineering/architecture design tool". The placement of objects was found to be slow by the participants. This can be improved in the future, for example by using 3D objects with fewer polygons. Several times the participants asked for an application example to better understand the approach. This is now available and is described in Section 5.2.3. It was also suggested that the description of use and damage scenarios could be made more precise by using more standard parameters, e.g. by specifying weather or speed (cf. Section 6.5.1 for the comprehensive evaluation of the approach).

5.3 Data-driven risk assessment in workshops

In this section I present the contents of a scientific paper [**JKA+22**]. This paper has two parts: I will present an approach for the assessment of damage scenarios (C_1). For this purpose, data from the German Federal Statistical Office (StBA) are used, which contain more than 10 million registered traffic accidents on German roads. The data covers the period 2017-2020. I have analysed the StBA data and evaluated it according to its usefulness for workshops. The result of this work is a set of concrete tables covering different accident types and causes. In two workshops with 17 experts from the automotive industry and experts from product development, the developed tables were used to evaluate different damage scenarios (C_2).

5.3.1 Need to use statistics at concept phase

During the concept phase, different use cases for the system to be developed are identified. The identification of such use cases takes place in workshops with various stakeholders, usually in a leading position [**JAD21**; **GRS14**]. The ISO/SAE 21434 standard [**ISO21**], which is relevant to the automotive industry, also requires the identification and evaluation of damage scenarios as part of the concept phase.

In this context, ISO/SAE 21434 recommends the use of the Automotive Safety Integrity Level (ASIL) [**ISO18**] risk classification scheme established in the automotive industry. In order to derive an ASIL that describes the risk of a damage scenario, various parameter values must first be determined. These parameter values are determined by the stakeholders. In particular, stakeholders must estimate by how often the damage scenario being considered occurs. The determination of these parameters is based on the experience of the stakeholders. In Germany, the Federal Statistical Office's road accident statistics record all police-documented accidents resulting in personal injury or property damage on public roads and places in millions of cases [**FSO21-ol**]. Validation of risk assessment based on stakeholder experience through statistical data currently occurs, if at all, in subsequent development steps rather than in the conceptual phase. However, in some cases a risk assessment based on stakeholder experience may differ significantly from a risk assessment based on statistical data. This can lead to mis-prioritisation at the beginning of product development, with the result that the mis-prioritisation is not recognised until later stages of product development. This leads to high costs in product development, as changes have to be communicated and implemented across several stages of product development.

By using statistical data as part of the ASIL rating, stakeholders can better estimate how often damage scenarios can occur. Since multi-stakeholder workshops are expensive to run and managers have little time for one-on-one meetings, adequate solutions need to be quick and easy to apply. In my analysis of 19 approaches (cf. Section 5.3.2), I could not find any approach that uses comprehensive statistical data in relation to the ASIL classification scheme, so that it can be applied in concept phase workshops.

I present the literature review in Section 5.3.2. In Section 5.3.3 I describe how I aggregated the statistical data and how I mapped this data to the ASIL risk classification scheme. In Section 5.3.4 I describe an approach to using the aggregated data for risk analysis. A summary of the evaluation and an outlook for future work is given in Section 5.3.5. A detailed evaluation is described in Section 6.5.2.

5.3.2 Related approaches using statistical data

In this section I present the approaches I have analysed (cf. Figure 5-6). I have considered approaches that are based on as much data as possible. In particular, I have examined whether these approaches support the evaluation of damage scenarios.

In the following I present the requirements according to which I evaluated the approaches:

R1: The information provided must be usable in a workshop with multiple experts from different domains. R1 is conditionally met if the information is difficult to understand and more in-depth statistical knowledge is required to understand and use it. *R2: The information provided must be empirically validated*. Based on the underlying number of traffic accidents of the investigated approaches, the following three clusters result: R2 is fulfilled if the number of underlying traffic accidents `n >= 10000`. R2 is conditionally fulfilled if `n < 2000`. Otherwise R2 is not fulfilled. *R3: The underlying data has to be up-to-date*. Here I assume that technical solutions already exist for frequently occurring traffic accidents from the past. R3 is fulfilled if the majority of the data is not older than 5 years. R2 is conditionally fulfilled if the majority of the data is not older than 10 years. R3 is not fulfilled if the majority of the data is older than 10 years. *Using R4-R6, I require that the provided information of the analyzed approaches must be able to be used for risk assessment according to the ASIL classification scheme*.

The parameters exposure, severity and controllability are used to determine the ASIL level. The ASIL level represents the risk for a damage scenario. An approach fulfils R4/R5/R6 if the parameter value can be obtained directly from the information provided. An approach conditionally fulfils R4/R5/R6 if the parameter value can be determined with additional effort from the information provided. An approach does not fulfil R4/R5/R6 if the parameter value cannot be determined because no data are available.

I describe the assessment using the following approaches as examples. The approach [**KBD+17**] is designed for use in workshops and provides several compact and easy to understand tables (R1 fulfilled). Extensive statistics have not been considered (R2 not met). The tables are based on approaches from 1974-2010 (R3 not met). The tables presented explicitly provide the parameter values for severity (R5 met) and controllability (R6 met), but there is no information on exposure (R4 not met). The [**KC21**] approach provides several tables, some of which require a deeper knowledge of statistics (R1 partially satisfied). The information is based on comprehensive and recent data from 2017-2018 on road accidents in the London region (R2 and R3 fulfilled). The exposure values of the damage scenarios considered can be obtained directly (R4 fulfilled). Damage effects must be determined individually for each table through intensive analysis of the work (R5 partially met). No information on controllability is available (R6 not satisfied). Overall, none of the approaches considered, and no trivial combination of approaches, meets all the requirements.

5.3.3 Data aggregation approach

As a basis for the approach, I examined all 16 tables of the German Federal Statistical Office on road traffic accidents. The tables are based on the data available at that time for the period 2017-2020. Based on the procedure described below, this results in the tables shown in Figures 5-8 - 5-10. I have chosen this source because it provides data on over 10.2 million road traffic accidents registered by the police during the period mentioned

R1	Information usable in workshop
R2	Information based on large-scale data
R3	Information based on current data
R4	Considers ASIL parameter Exposure
R5	Considers ASIL parameter Severity
R6	Considers ASIL parameter Controllability

To what extent do the approaches considered satisfy the requirements?

Satisfied | Partially s. | Not satisfied

		Considered appraoches	Requirements R1	R2	R3	R4	R5	R6
n<10	[KDA+20]	ASIL estimation using fuzzy set theory						
	[KBD+17]	Objectification of HARA						
	[SV20]	Simulation-based methodology for HARA						
n<2000	[CGA20]	Analysis of the risk factors						
	[SS17]	Crash severitiy factors						
	[KSM+20]	Risk factors contributing to road traffic accidents						
	[WLS+20]	Risk factors on victims of traffic accidents						
	[Cai20]	Cause analysis of traffic accidents						
n>10000	[OG18]	Risk factors on occurrence time of traffic accidents						
	[JSB+20]	Road trafc crashes in Ilam province						
	[KC21]	Hazard-based model of traffic incident durations						
	[CS14]	Risk factors affecting raod traffic accidents						
	[AFB+13]	Risk on severity of traffic accidents						
	[AHS+19]	Factors involved in road accidents						
	[LPP+20]	Road traffic accidents on Lithuanian roads						
	[LCZ+18]	Road accident statistical annual report of China						
	[LCZ+18]	Road traffic accidents in India						

Figure 5-6: Evaluation of related approaches.

[**FSO21-ol**].

Each table contains data on related causes of accidents. Each row in a table represents a specific cause of an accident, e.g. turning mistake. Some of the 16 tables dealt with different aspects of an identical group of accident causes. This allowed an additional aggregation of several tables into one table. To aggregate the data from the 16 tables, I proceeded as follows for each table: Step 1: Calculate the sum of the number n of road accidents in the period considered. Step 2 & 3: Sorting the rows in ascending order. The rows were aggregated into 4 groups so that they could be assigned to exposure levels E1-E4. The median was used as a separator for E1,E2 and E3,E4. Step 4: Add columns S1-S3 (severity levels S1-S3) for each row if data are available for the corresponding columns. For S1 and S2 this is the case for all tables. For S3 this is the case for Tables 6, 9 and 13. For Tables 9 and 11 additional data were available on the distribution of accident causes by severity S1-S3. Step 5: Columns C1-C3 (controllability level C1-C3) were added for each row. This allows an assessment of how easy or difficult it is for the driver to react to the considered damage scenario in order to avoid it. Step 6: Calculate the ASIL levels for each entry in the table according to the ASIL classification scheme.

Question-based selection of relevant tables			Relevance	Relevant entry from the table	Exposure	Severity	Controllability	ASIL
Can the cause of the accident be the driver?	Is it due to driver misbehavior?	1	Yes	Insufficient distance, turning mistakes	E4	S2	C2	B
What non-driver-related causes can play a role as the cause of the accident?	Do technical defects play a role?	2	No					
	Are obstacles, weather or road conditions relevant?	3	Yes	Obstacles on/near street	E2	S2	C2	QM
	Is it due to cyclists misbehavior?	4	No					
	Is it due to pedestrian misbehavior?	5						
What other aspects can play a role as the cause of the accident?	Do I know the road users who suffer injury?	6	Yes	Persons in the vehicle	E4	S3	C2	C
	Is the type of road user of the main cause of the accident known?	7		Persons in the vehicle	E4	S2	C2	B
	Are types of road users known who have suffered damage as a result of the accident?	8		Persons in the vehicle	E4	S2	C2	B
	Are bicyclists or e-scooter users involved in the accident?	9	No					
	Is the damage location or damage type relevant?	10	Yes	Outside buid-up areas, Personal injury	E2	S2	C2	QM
	Is it only property damage or personal injury as well?	11		Personal injury	E3	S3	C2	B
	Does year period matter?	12	No					
	Does the age of the persons involved matter?	13						

Figure 5-7: Result of applying the approach in a workshop with 10 product development experts.

5.3.4 Risk assessment based on statistical data

In this section, I present an approach for assessing damage scenarios in workshops. I illustrate the approach with the results of a workshop. This workshop was attended by 10 experts in the field of product development (cf. Section 6.5.2). The approach consists of a method (cf. Figure 5-7) and 13 tables (cf. Figures 5-8 - 5-10) for risk assessment based on comprehensive statistical data (n=10 million).

The approach consists of four steps. Step 1: Identification of damage scenarios for the system to be developed and initial prioritisation. The damage scenario is visualised and the process of the damage scenario is discussed. In the workshop the participants decided on the damage scenario *overtaking vehicle not visible*. 3D Engineer [**JSK+22**] was used to visualise the damage scenarios. Step 2: Based on the discussed damage scenario, the exact description is documented. In this case: *Entering a road is critical if there is a vehicle on the opposite road and the view of the road to be entered is blocked by an obstacle. Behind*

	%	Description (n = 1,4 million)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	4	Overtaking mistakes	Q	Q	Q	Q	Q	Q
	4	Influence of alcohol						
	4	Improper behavior towards pedestrians						
E2	7	Improper road use	Q	Q	Q	Q	Q	A
E3	12	Inappropriate speed	Q	Q	A	Q	A	B
E4	14	Insufficient distant	Q	A	B	A	B	C
	14	Failure to yield right of way						
	16	Turning off mistakes, mistake in starting off or entering the road from premises et cetera						

Table 1: Driver-related causes of accidents involving personal injury

	%	Description (n = 14 thousand)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	2	Towing equipment	Q	Q	Q	Q	Q	Q
	5	Steering mechanism						
E2	16	Lighting	Q	Q	Q	Q	Q	A
E3	18	Brakes	Q	Q	A	Q	A	B
E4	29	Tyres	Q	A	B	A	B	C

Table 2: Causes of accidents involving personal injury - Technical faults

	%	Description (n = 120 thousand)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	0,4	Obstacles: Road construction site on carriageway not or not sufficiently secured	Q	Q	Q	Q	Q	Q
	1	Influence of the weather: Obstruction of visibility by fog						
	3	Obstacles: Other animals on the carriageway						
E2	8	Obstacles: Wild animals on the carriageway	Q	Q	Q	Q	Q	A
E3	11	Influence of the weather: Obstruction of visibility by dazzling sunshine	Q	Q	A	Q	A	B
	14	Road surface conditions: Slippery carriageway by snow, ice						
E4	18	Road surface conditions: Slippery carriageway by rain	Q	A	B	A	B	C

Table 3: General causes of accidents involving personal injury

Figure 5-8: A - Statistically validated ASIL tables for workshop use.

	%	Description (n = 65 thousand)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	2	Improper behavior towards pedestrians	Q	Q	Q	Q	Q	Q
	3	Overtaking mistakes						
	4	Insufficient distant						
E2	7	Influence of alcohol	Q	Q	Q	Q	Q	A
	7	Failure to yield right of way						
E3	8	Turning off mistakes, mistake in starting off or entering the road from premises et cetera	Q	Q	A	Q	A	B
	10	Inappropriate speed						
E4	17	Improper road use	Q	A	B	A	B	C

Table 4: Causes of accidents involving personal injury: improper driving of cyclists

	%	Description (n = 49 thousand)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	0,4	On pedestrian crossing without control by policeman or traffic lights	Q	Q	Q	Q	Q	Q
	2	Failure to use footway						
	5	Near junctions, traffic lights or pedestrian crossings with heavy traffic at other places						
E2	8	At places where the pedestrian traffic was controlled by policeman or traffic lights	Q	Q	Q	Q	Q	A
E3	14	By suddenly emerging from behind obstacles obstructing the visibility	Q	Q	A	Q	A	B
E4	46	Without paying attention to the traffic	Q	A	B	A	B	C
	79	Improper behaviour when crossing the carriageway						

Table 5: Causes of accidents involving personal injury - Improper behaviour of pedestrians

	%	Description (n = 1,2 million)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	1	Drivers of buses and coaches	Q	Q	Q	Q	Q	Q
	2	Pedestrians						
	2	Drivers of motorcycles with insurance sign						
E2	5	Drivers of motorcycles with official sign	Q	Q	Q	Q	Q	A
E3	6	Drivers of goods road motor vehicles	Q	Q	A	Q	A	B
E4	15	Drivers of bicycles	Q	A	B	A	B	C
	16	Drivers of passenger cars						

Table 7: Main accident perpetrator in accidents causing personal injury

Figure 5-9: B - Statistically validated ASIL tables for workshop use.

	%	Description (n = 1,5 million)	S1			S2			S3		
			C1	C2	C3	C1	C2	C3	C1	C2	C3
E1	2	Buses and coaches	Q	Q	Q	Q	Q	Q	Q	Q	A
	2	Goods road motor vehicles									
	4	Motorcycles with insurance sign									
E2	8	Motorcycles with official sign	Q	Q	Q	Q	Q	A	Q	A	B
	8	Pedestrians									
E3	23	Bicycles	Q	Q	A	Q	A	B	A	B	C
E4	54	Passenger cars	Q	A	B	A	B	C	B	C	D

Table 6: Persons injured/killed in traffic accidents, by type of traffic participation

	%	Description (n = 2,2 million)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	1	Drivers of Buses and coaches	Q	Q	Q	Q	Q	Q
	2	Drivers of Motorcycles with insurance sign						
E2	5	Drivers of Motorcycles with official sign	Q	Q	Q	Q	Q	A
	5	Drivers of Goods road motor vehicles						
	6	Pedestrians						
E3	17	Drivers of Bicycles	Q	Q	A	Q	A	B
E4	62	Drivers of Passenger cars	Q	A	B	A	B	C

Table 8: Persons involved in accidents causing personal injury

	%	Description (n = 42 thousand)	S1 (82%)			S2 (17%)			S3 (1%)		
			C1	C2	C3	C1	C2	C3	C1	C2	C3
E2	1	E-Scooters	Q	Q	Q	Q	Q	A	Q	A	B
E4	20	Bicycles	Q	A	B	A	B	C	B	C	D

Table 9: Traffic accidents of e-scooters - A comparison

	%	Description (n = 10,2 million)	S1			S2		
			C1	C2	C3	C1	C2	C3
E1	1	Personal injury: On motorways	Q	Q	Q	Q	Q	Q
E2	2	Personal injury: Outside built-up areas, excl. motorways	Q	Q	Q	Q	Q	A
E3	8	Personal injury: Within built-up areas	Q	Q	A	Q	A	B
E4	88	Accidents involving material damage only	Q	A	B	A	B	C

Table 10: Accidents registered by the police by type of damage/location

Figure 5-10: B - Statistically validated ASIL tables for workshop use.

	%	Description (n = 10,2 million)	S1 (82%)			S2 (17%)			S3 (1%)		
			C1	C2	C3	C1	C2	C3	C1	C2	C3
E1	1	Serious accidents involving material damage: under the influence of intoxicating substances	Q	Q	Q	Q	Q	Q	Q	Q	A
E2	3	Serious accidents involving material damage: in the strict sense	Q	Q	Q	Q	Q	A	Q	A	B
E3	12	Accidents involving personal injury	Q	Q	A	Q	A	B	A	B	C
E4	85	Other accidents involving material damage	Q	A	B	A	B	C	B	C	D

Table 11: Accidents registered by the police: specification - Accidents and casualties

	%	Description (n = 12,2 thousand)	S3		
			C1	C2	C3
E1	13	February - March	Q	Q	A
E2	22	November - January	Q	A	B
E3	26	May-April, October	A	B	C
E4	39	June - September	B	C	D

Table 12: Persons killed in road traffic accidents, by month

	%	Description (n = 1,5 million)	S1			S2			S3		
			C1	C2	C3	C1	C2	C3	C1	C2	C3
E1	5	15 to 18	Q	Q	Q	Q	Q	Q	Q	Q	A
E2	7	under 15	Q	Q	Q	Q	Q	A	Q	A	B
E3	13	65 and over	Q	Q	A	Q	A	B	A	B	C
	16	18 to 25									
E4	59	25 to 65	Q	A	B	A	B	C	B	C	D

Table 13: Persons injured/killed in traffic accidents, by age

Figure 5-11: B - Statistically validated ASIL tables for workshop use.

the obstacle could be an overtaking vehicle. Exiting the vehicle could lead to an accident.
Step 3: In this step, the risk assessment is carried out based on statistical data. To save time in the workshop, guiding questions were formulated to help workshop participants identify relevant statistical data for the damage scenario under consideration (cf. Figure 5-7). In order to provide the workshop participants with a simple decision support tool for risk assessment, the statistical data were aggregated in the form of 13 tables and combined with the ASIL risk classification scheme (cf. Figures 5-8 - 5-11). In Section 5.3.3 I describe how I have approached the aggregation of the statistical data. In the application example, the tables to be considered could be reduced to the following: Tables 1,3,6,7,8,10 and 11.

In the following I will explain the breakdown of registered accidents using Table 1 as an example. In total, n=1.4 million registered accidents of the category *Driver-related causes of accidents with personal injury* were recorded by the police in Germany in the period 2017-2020. The table shows the percentage distribution of the related subcategories. For example, 4% of the 1.4 million accidents were caused by *Mistake*[1]

I illustrate the further procedure with the help of Table 1, which describes *Driver-related causes of accidents involving personal injury*. For each table, the relevant entries for the damage scenario are identified. If more than one entry is relevant, the most critical value is used. The entries *Insufficient distance* and *Turning mistake* in Table 1, both in category E4, were identified as relevant by the participants. In general, relatively infrequent cases are assigned to E1 and relatively frequent cases are assigned to E4. Participants then select the severity level of the damage scenario. S1 describes slight injuries, S2 severe injuries and S3 fatal injuries. The participants chose S2. The next step is to determine how well the driver can react to the damage scenario in order to avoid it. C1/C2/C3 means that it is easy/medium/difficult for the driver to avoid the damage scenario. The participants chose C2. Using the determined parameter values, the ASIL level for the considered damage scenario can be determined. There are values Q (quality management), A, B, C and D. Q represents the lowest risk, while D represents the highest risk. In the application example, the ASIL level for Table 1 is B for the determined parameters E4, S2 and C2. This procedure is repeated for the other tables identified as relevant in Step 2. This is done to consider further aspects of the damage scenario in the context of frequently occurring cases. The result is a set of individual ASIL levels. In the application example: B,Q,C,B,Q,B

Step 4: A representative ASIL level is added to the damage scenario description. The value is determined by taking the median of the determined ASIL values. This is done as follows: The ASIL values are assigned to the numbers 0 for Q to 4 for D and the median of these numbers is determined. The retranslation is then performed. The application example is 0(Q),0(Q),2(B),2(B),3(C). The median is 2(B), so the representative ASIL value for the

[1] The sum of the percentages does not add up to 100% in every table. The reason for this is that for some accidents there is a general assignment to a category (table), but no information about an assignment to a subcategory. Since no relative probabilities are available for such cases, they can be grouped together in E4 as a precaution.

considered damage scenario is B.

5.3.5 Evaluation summary and identified limitations

In various contexts, I have spoken to experts in the automotive sector. One result is that risk analysis in the concept phase is based on empirical knowledge and "gut feeling". In general, all experts were interested in easily applicable, statistically validated resources for risk assessment in the concept phase. I analysed data from the Federal Statistical Office on traffic accidents in Germany, aggregated them and mapped them to the ASIL risk classification scheme in the form of 13 tables. In addition, I developed an approach to use the tables based on identified damage scenarios. The approach was first used with students to get initial feedback and make necessary corrections. The approach was then used in two workshops with a total of 17 subject matter experts. In general, the approach was well received by the subject matter experts. The approach was described as intuitive and structured, which could be applied without prior knowledge. It was noted that the task description was unclear. This was due to the fact that at this point the approach was mostly communicated verbally and there was no example of its use. This has been corrected in the Section 5.3.4. It was unclear what happens after the damage scenarios have been evaluated (cf. Section 6.5.2 for the comprehensive evaluation of the approach). The relationship of the approach to the risk analysis steps of ISO/SAE 21434 needs to be better described in the future. Furthermore, it was unclear how exactly the mapping between the considered damage scenarios and the scenario descriptions in the tables should be performed. This needs to be more clearly described in the future.

5.4 Model transformation

In this section, I present the results of five research-related industry projects. The customer was a German premium vehicle manufacturer. My role was to elicit the customer's requirements and to plan and manage the development activities. This work has two contributions. In the first project, I analysed a company standard for its compliance with common Model-Based Systems Engineering (MBSE) approaches [**OEM19b**]. The company standard describes how a general system architecture can be created during the concept phase. The company standard defines the Effect Chain Modelling Language (ECML) as the modelling language. This modelling language has been used in practice in the company for more than 10 years. In order to be able to use the general system architecture in the detailed system design, a model transformation to the Systems Modelling Language (SysML) is necessary. SysML is the de facto modelling language in MBSE [**DOR16**] and is used by the customer for detailed system design. For this purpose, a software prototype (C_1) has been developed in three completed projects and one initiated project, which enables model transformation with operational data [**OEM20**; **OEM21a**; **OEM22a**; **OEM22c**]). In the context of these projects, the prototype has been evaluated

on the basis of several cases (C_2).[2]

In Section 5.4.1 I explain the background to the use of ECML. In Section 5.4.2 I first introduce the mapping between ECML and SysML. Then, in Section 5.4.3, I illustrate the mapping with an example. A summary of the evaluation and an outlook on future work is given in Section 5.4.6. A detailed evaluation is described in Section 6.5.3.

5.4.1 Need for the use of the ECML

The Effect Chain Modelling consists of the ECML and a method for creating a system architecture. Effect Chain Modelling is a creative method. Due to its relative simplicity, it can be understood and applied in offline and online workshops with several domain experts without intensive training. The creation of a system architecture is done in the client company using Microsoft Visio. Digital templates representing the graphical constructs of the ECML are used. In the workshop, the moderation and modelling is mainly done by a systems engineering expert. The domain experts provide input for the modelling. The domain experts also check that the model is correct.

The use of Microsoft Visio simplifies usability and does not restrict the creative flow of workshop participants through complex model constructs. Because the ECML uses a simple color scheme, model constructs can be quickly distinguished visually. Unlike traditional brown paper modelling, it is easier to make changes in digital form. In addition, the problem of illegible handwriting is eliminated as text can be entered via text boxes using the keyboard. Unlike brown-paper modelling, a digital system architecture is not limited in the number of elements it contains or its size. In addition, digital model elements can be rearranged more easily, allowing implicit relationships to become apparent quickly.

SysML is a standardised modelling language with diagrams for modelling structure, behaviour and requirements. There are 9 diagram types in total. SysML tools such as Cameo Systems Modeler make it easy to manage the complexity of complex models and comply to the extensive syntactic rules of SysML. The SysML specification is detailed in over 340 pages [**SYS15**]. The SysML specification is based on the UML specification, which is 730 pages long [**UML17**].

Understanding individual SysML diagrams, which do not use complicated model constructs, requires only simple instruction. Proper modelling with SysML requires training. There are different types of blocks, ports and relationships. Without expertise, consistent modelling is not possible. In addition, this would lead to a high level of coordination and editing in engineering. Due to the high entry threshold, SysML-compliant modelling is not suitable as a creative method in the context of interdisciplinary design during the concept phase.

---

[2] In order to maintain confidentiality, only the general approach is explained in the following, based on publicly available work and illustrated with an own example.

5.4.2 Mapping ECML to SysML

In order to be able to use the system architecture created in the concept phase in the system design phase, it is necessary to transform the ECML model into SysML. In the following, I present the elaborated mapping between the two modelling languages (cf. Figure 5-12). The mapping is based on numerous consultations with the customer company.

ECML consists of three types of graphical elements: Blocks, Interactions and Effect Types. In general, all ECML elements are mapped to SysML Internal Block Diagram (IBD) elements. An interaction is a relationship between two ECML ports. Interactions are differentiated as follows Intentional, Unintentional and Misuse. Interactions are mapped to SysML connectors. An appropriate stereotype is used to distinguish the type of interaction. A connector connects two SysML ports. ECML allows different port types to be distinguished using effect types. Figure 5-12 lists some effect types. For example, the Information Effect Type or the Mechanical Effect Type. The effect types are mapped to SysML interface blocks. Interface blocks allow the typing of ports in SysML.

Figure 5-13 shows an ECML2SysML profile. This profile extends SysML with the interaction and effect types of ECML using stereotypes. To represent the different ECML blocks, SysML blocks and SysML part properties are used (cf. Figure 5-12).

5.4.3 Explanation of the ECML to SysML mapping using an example

The Figure 5-14 shows an example of a mapping between an ECML model and a SysML model. The model is a system architecture for the realisation of the User Story *Warn Driver (The driver's steering wheel vibrates if an obstacle is detected by the platoon leader's vehicle)*. See Section 5.5 for a more detailed description of the example and the meaning of the model elements.

In general, all ECML blocks are mapped to SysML blocks or SysML part properties.

ECML Interactions are associated with ECML Ports. Where ECML Ports have a Port Direction and an Effect Type. ECML Interactions and their labels are mapped to SysML Connectors. In the following, compare the ECML interaction *Sensor data* between the ECML blocks *Obstacle* and *Multi purpose camera*. The *Obstacle* ECML block represents an untyped ECML block and therefore, in particular, does not represent a system, component or element. The ECML block *Multi purpose camera* represents a component. A *Multi purpose camera* can detect an *Obstacle*, therefore there is an information relationship between the *Obstacle* and the *Multi purpose camera*, called *Sensor data*. The information relationship is represented in ECML by the effect type named *Information* using an icon as part of the associated ECML ports.

ECML ports, including effect type and port direction, are mapped to SysML ports in several steps. Step 1: Depending on the concrete effect type, a mapping to a corresponding SysML interface block is performed first. Interface blocks allow a detailed specification of

Elements of the Effect Chain Modeling Language (ECML)	Description of the elements of the Effect Chain Modeling Language	Mapping to elements of the Systems Modeling Language (SysML)	Mapping description
Item boundary ABC	With the help of the „Item boundary“, the system under consideration is clearly delimited from its operational environment.	«Block» ABC	Mapping of the element "Item boundary" of the ECML to the element «Block» of the SysML IBD diagram
ABC	Representation of untyped elements.	«Part Property» ABC	Mapping of an untyped element of the ECML to the element «Part Property» of the SysML IBD diagram.
System ABC	Representation of the „System“ at the system environment level. It corresponds to the product to be developed.	«Block» ABC	Mapping of the element "System" of the ECML to the element «Block» of the SysML IBD diagram.
Component ABC	Representation of the system internal or external „Component“s.	«Part Property» ABC	Mapping of the element "Component" of the ECML to the element «Part Property» of the SysML IBD diagram.
Element ABC	Representation of the system internal or external „Element“s..	«Part Property» ABC	Mapping of the element "Element" of the ECML to the element «Part Property» of the SysML IBD diagram.
Description D Description D Description D	„Intentional/Unintentional/ Misuse Interaction“ Interface between two systems/ components/elements.	:X Description D :~X «intentional» :X Description D :~X «unintentional» :X Description D :~X «misuse»	Mapping of the "Intentional/ Unintentional/Misuse Interaction" element of the ECML to the connector between two ports of the SysML IBD diagram with additional use of the «intentional/unintentional/ misuse» stereoptype. Typing of the associated ports of type X. Conjugation of the incoming port.
Mechanical Information / Software Substance / Material Electrics / Electronics Power / High voltage Thermal energy Light / Optics Airborne sound	Concretization of the interaction between two systems/components/elements using the "Mechanical/..." effect types.	:Mechanical IF :Information / Software IF :Substance / Material IF :Electrics / Electronics IF :Power / High voltage IF :Thermal energy IF :Light / Optics IF :Airborne sound IF	The effect type "Mechanical/..." of the ECML is mapped to an Interface Block of the SysML with the name "Mechanical IF/...". This allows the concretization of SysML Proxy Ports to the type "Mechanical IF".

Figure 5-12: Mapping ECML to SysML.

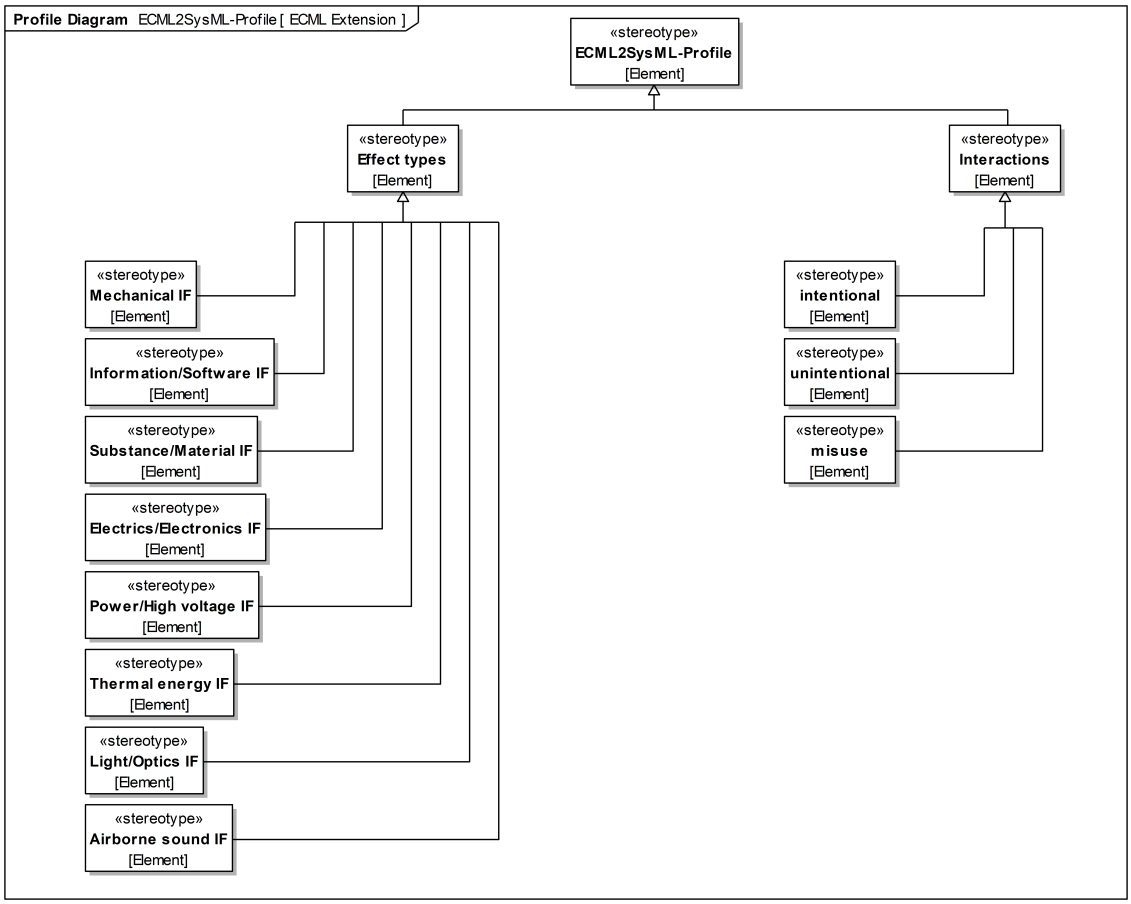


Figure 5-13: SysML2ECML profile.

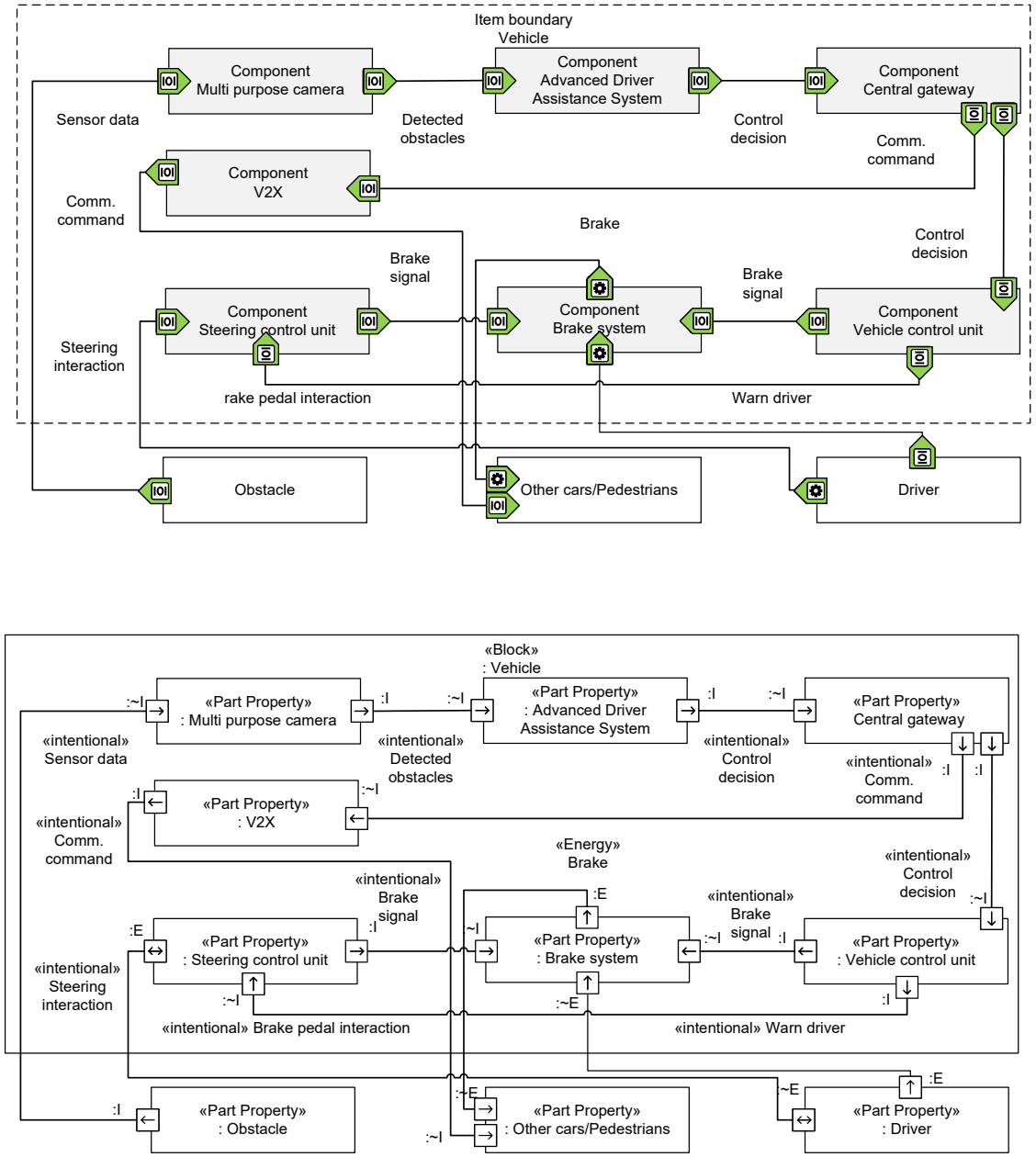


Figure 5-14: Illustration of the ECML to SysML mapping.

SysML ports. This allows the precise specification of complex ports with numerous flow properties. A flow property consists of a name, a type and a flow direction. This is also the required way to describe SysML ports since the SysML 1.4 specification (cf. [**SYS15**]). In the example, the effect type *Information* is mapped to a SysML interface block named *Information*. For ease of reading, I use the short form *I* instead of *Information* in the figure.

Step 2: Mapping the port direction of the associated ECML blocks to the flow direction of the associated SysML interface blocks. Step 3: Mapping of interface blocks to SysML ports. SysML has the concept of conjugated ports. Conjugated ports are represented by the prefix $\sim$. There is a reference relationship between the set of flow properties of the associated interface block and the set of flow properties of the conjugated port. The difference between the two sets of flow properties is that the flow directions of conjugated ports are reversed. The advantage of conjugated ports is that changes to the port specification are propagated to the conjugated port and the flow directions are automatically inverted. The connector named *Sensor data* is conjugated to an outgoing port of the SysML element *Obstacle*. The port type is *Information (I)*. The same connector is also connected to an incoming port of the SysML element *Multi purpose camera*. Also of port type *Information (I)*. If you did not use a conjugated port on the *Multi purpose camera* SysML element for the connector in question, the port direction would be outgoing and therefore the wrong way round.

5.4.4 Requirements for the prototype

In this section I present the requirements for the prototype. These requirements are based on numerous consultations with the customer company over a period of 3 years and on an analysis of ECML system architectures provided by the customer company during this period.

Fundamentally, Microsoft Visio is a visualisation tool and not a MBSE tool. For this reason, an ECML system architecture created in Visio is programmatically just a set of templates that are aligned and superimposed, each representing different ECML elements. This means that in most cases there is no clear assignment and no real link between the stencils (or ECML elements) that can be recognised by the programming libraries. In this context, the following requirements emerged

R1: Extraction of non-hierarchical ECML blocks. *R2: Extraction of ECML ports*. Visually, an ECML port is associated with an ECML block. To be precise, there is only an overlay between the two elements. An appropriate function is required to determine and store a mapping of the ECML port to the ECML block. This function must be able to detect an overlay of these two elements and exclude an overlay on other elements. *R3: Extraction and mapping of effect types to ECML ports*. This requires an overlay function as in *R2*. *R4: Extraction of explicitly linked interactions*. Interactions in ECML are realised by connecting lines in Visio. To extract these interactions, the connectors must be explicitly linked to ECML ports.

R5: Extraction of hierarchised ECML blocks. Hierarchised ECML blocks are visually nested. Programmatically they are only superimposed. A suitable overlay function is required to ensure that the nesting is recognised. *R6: Extraction of the flow direction of ECML ports*. In this case, a function is needed to determine whether the port is outgoing, incoming or unidirectional based on the port orientation. *R7: Extraction of implicitly linked interactions*. In order to reduce the modelling effort, explicit linking of interactions to ECML ports is no longer assumed. An appropriate overlay function must be provided. *R8: Plausibility check of ECML models*. Since an ECML model can have many blocks, ports and relationships, there is a risk of undetected modelling errors. An example of this is when an interaction is linked to only one ECML port. To ensure that modelling errors are detected early and not carried over into the SysML model, a plausibility check is required.

5.4.5 Description of the implemented prototype

In general, the prototype consists of two separate tools connected by a common exchange format. The first tool is the *Visio Exporter*. The tool reads Visio files containing ECML models and extracts this information into an exchange format. The *Visio Exporter* was developed because ECML models cannot be exported from Microsoft Visio to an exchange format using the tools available at the time of development.

The tool was developed using a *.Net* library in the *C-Sharp* programming language. This library comes from Microsoft and was used to programmatically process Visio files.

JSON was chosen as the exchange format [**Jso23-ol**]. JSON is a compact file format that has a simple and readable text form and is used as a data exchange format. In particular, JSON is used to store structured data. Since ECML models represent structured data, JSON is an appropriate data format. The *Visio Exporter* extracts only the data required for model generation. In general, Visio files available in XML format could have been used directly. However, such files usually contain information that is irrelevant to the generation of a SysML model. For example, a file containing a single ECML element has a length of several 1000 lines, but only a few lines are relevant for model generation.

Based on ECML models in JSON format, it is possible to implement import tools for different SysML tools. Cameo Systems Modeler (CMS) is used by the client company for detailed system design. CMS is an industry-leading MBSE environment that provides the most standards-compliant SysML models and diagrams. The second tool is the *Cameo Importer*. This tool has been developed as a plug-in for CMS. The plug-in generates a SysML model from a JSON file with an ECML model following the mapping presented in Section 5.4.2. The plugin is written in the *Java* programming language and uses the CMS API.

5.4.6 Evaluation summary and identified limitations

The prototype was developed based on customer requirements that were regularly elicited during 5 research-related industrial projects and evaluated through several use cases. The first step was to investigate whether an existing commercial plug-in would be sufficient for the company's use cases. As the commercial plugin only provides limited support for the language elements used in ECML and as the plugin only supports a limited character set, the company preferred to develop a new plugin. The prototype was developed incrementally and iteratively. First the customer requirements were identified and agreed. Then the prototype was developed or refined and tested. Then the prototype was presented and additional customer requirements for further development were identified and agreed (cf. Section 6.5.3 for the full evaluation of the approach). The developed tool transforms ECML models to SysML models. The tool does not yet take into account the company-specific SysML language extensions at the vehicle manufacturer, so that there is not yet complete compatibility with the SysML models used in the company. Manual adjustments still have to be made in order to use the models in the company.

5.5 Threat identification in workshops

In this section I present the contents of a scientific paper [**JAD21**]. This paper has three contributions.

In this section I present a method to be used in workshops (C_1). The method supports the creation of an initial system model. The method is designed to be used by an interdisciplinary team of stakeholders. The system model serves as a communication tool between the stakeholders in the workshop. Based on the system model, I explain how security threats can be identified in the workshop. The method is illustrated with an example from the automotive sector. This should help the workshop participants to build a realistic system architecture, 69 descriptions of vehicle components from an automotive supplier were analysed, of which the most important are presented below (C_2). The approach was evaluated in workshops with students (C_3).

In Section 5.5.1 I explain the context and problem of the work. In Section 5.5.2 I present the literature review. In Section 5.5.3 I give an overview of the approach. In Sections 5.5.4 to 5.5.6 I present the individual steps of the approach. These are: Identification, refinement and prioritisation of threat scenarios. A summary of the evaluation and an outlook for future work is presented in Section 5.5.7. A detailed evaluation is described in Section 6.2.

5.5.1 Background and necessity of threat identification in workshops

Cyber-physical systems (CPS), such as autonomous vehicles, are intelligent and networked. The development of such systems and their components is characterised by close cooperation between mechanical, electrical and software engineering [**GRS14**]. The interdisciplinarity and complexity of these systems leads to an increasing challenge for effective

and efficient development. A lack of understanding of the system between stakeholders can result in security threats not being identified in the design phase of the system architecture. This can lead to high costs in subsequent phases of product development.

A lack of security in a CPS can compromise the security of the CPS and damage the company's reputation. In 2015, hackers demonstrated an attack on a moving SUV [**Gre15-ol**]. In this case, the infotainment system was remotely hacked, allowing the hackers to take control of the vehicle. This led to a recall of 1.4 million vehicles from the affected company [**Gol15-ol**].

To identify security threats early in the development process, it is necessary to take a holistic view of the system being designed. This involves involving multiple stakeholders from different disciplines, most of them are not familiar with security. Model-Based Systems Engineering (MBSE) improves the understanding of systems between stakeholders through the use of models. SysML is the de facto modelling language in MBSE [**DOR16**].

Based on the literature review in Section 5.5.2, I formulate the following research question: *How can an interdisciplinary team of stakeholders use SysML to identify safety-relevant security threats in a workshop, so that an initial input for the system design phase can be determined?*

5.5.2 Analysis of related approaches

I have selected the literature according to the following criteria: Number of citations, publications preferably from the last 10 years. Established security & safety approaches are partially applicable only to specific engineering disciplines, such as software engineering [**ML06**; **MS05**; **ISO18b**; **RDG+02**; **Fer13**; **MWZ19**]. Other approaches are applicable across disciplines at the system level, but only partially consider security threats & safety hazards in an integrated way. The SREP approach does not consider safety [**RAG18**], while the following approaches do not consider security [**ADK+20**; **ISO18**; **ISO15**; **Pol16**; **Rup14**]. The Cybersecurity Guideline for Cyber-Physical Vehicle Systems [**SAE16**] and the SAHARA approach [**MSB15**] consider security risks. Unfortunately, the Cybersecurity Guideline and SAHARA do not use models. Approaches such as [**CDP+19**; **THZ17**] use models but do not use SysML. Approaches such as CONSENS 3D [**JKK20**] and Security by MBRE [**Jap20**] use SysML in the context of MBSE. However, a concrete method for identifying safety-relevant security threats is missing.

5.5.3 Overview of the method and introduction of the application example

The method consists of three steps: Step 1: Creation of a black box model to determine the overall system boundary and identify security threats at the system boundary. Step 2: Building a white box model. This involves deriving an initial high-level system architecture based on the black-box model and refining the identified threats. Step 3: Prioritization of the identified threats within the initial high-level system architecture.

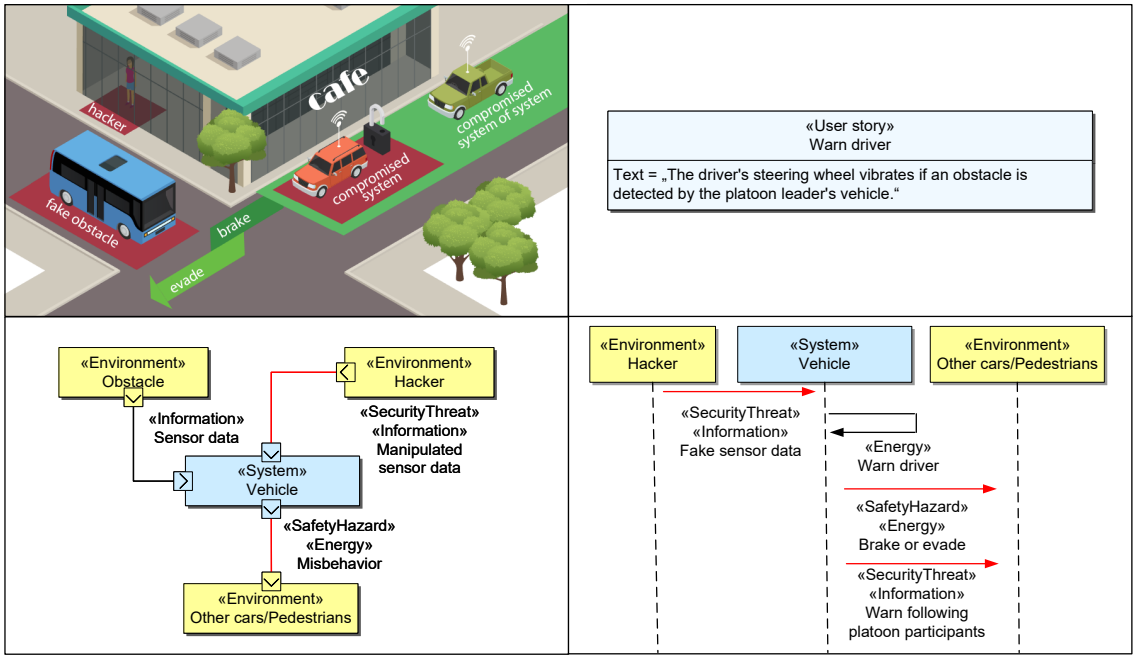


Figure 5-15: Black box threat identification.

I will illustrate this approach with the example of platooning. Platooning describes the networking of several vehicles which, with the help of a technical control system, can drive behind each other at very short distances without compromising road safety. Figure 5-15 shows a platooning situation: The platoon reaches an intersection. The platoon leader detects an obstacle. Here represented by a bus. A hacker could be sitting in the café. The platoon leader's vehicle can brake, evade, do nothing and warn the platoon members. Different user stories can belong to the sketched situation. In the following I illustrate the approach with the user story "Warn driver" (cf. Figure 5-15).

5.5.4 Identify threats at the system boundary

In Step 1 I explain how the black box model is created (cf. Figure 5-15). This includes a simple visualisation of the initial situation, the derivation of user stories, the creation of the environment model and the definition of use cases and threat cases.

Visualisation is an easy way to ensure a common understanding between the different stakeholders. Visualisation can be done in different ways. For example, drawing on a (digital) whiteboard or using a 3D environment such as 3DE (cf. Section 5.2) to visualise the initial situation using 3D objects. User stories are derived from the visualised initial situation. In the context of MBSE these are a general form of system requirements. In the context of SysML, user stories are assigned to the Requirements diagram type. Figure 5-15 shows the *Warn driver* user story. It describes one aspect of the visualised initial situation from the user's point of view. In this case, the driver's point of view.

Based on the visualised initial situation and the set of user stories, the environmental model

is created. In the environment model the system to be designed is considered as a black box. Here the connection to other environmental objects is specified. The CONSENS stereotypes Information, Energy and Substance are used to concretise the environmental relationships (cf. Section 2.2.3). In the context of SysML, the environment model is mapped to the Internal Block Definition Diagram (IBD) structural model. To distinguish between security threats and security hazards, I have introduced appropriate stereotypes. Since the distinction between security threats and security hazards has always been a problem for participants in workshops held before [**RE20-DS**], I have introduced the following generally understandable definitions: (D1) A security threat exists in a particular element if hackers can exploit a vulnerability to gain unauthorised access to the system. (D2) A security vulnerability exists in a particular item if the system can cause physical damage to the system or its environment.

Based on the workshop conducted (cf. Section 6.2), I found that the use of these stereotypes already leads to first discussions between the participants about the classification of security and safety aspects. This allowed the first ambiguities to be identified and resolved.

5.5.5 White box threat refinement

In Step 2 of the approach, I explain how to derive a white box model from the black box model (cf. Figure 5-19). This includes identifying the necessary system components and relationships to build a high-level system architecture in the form of a SysML Internal Block Definition Diagram (IBD). This step also includes the concretisation of the threat cases using the identified necessary system components using SysML Sequence Diagrams.

In order to ensure a common understanding between different stakeholders, the 18 most important system components and their interrelationships for autonomous vehicles have been identified and summarised in Figures 5-16 - 5-18. The component table is based on an evaluation of 69 product descriptions of safety-relevant vehicle components from Bosch [**BPS20-ol**].

Based on the black-box sequence diagram in Figure 5-15, the white-box sequence diagram in Figure 5-19 is extended to include system-level functions and components. The components and architectural constraints from Figures 5-16 - 5-18 are used as support.

5.5.6 White-box threat prioritisation

In order to prioritise activities in the subsequent engineering phases, the identified security threats need to be evaluated (cf. Figure 5-19, F1-F13). For this purpose, the SAHARA method [**MSB15**] is used in combination with the ASIL risk classification scheme (cf. Section 2.3.1) for each function in a sequence diagram. It is important to note that defined functions and their associated ratings can be reused in other sequence diagrams.

The STRIDE approach [**Str19-ol**] is used to stimulate constructive discussion and to

	Name	Description	Functions	Architectual Constraints
Main components	Central Gateway	The central gateway is the central communication node, acts as a router (for in-vehicle communication and through the communication unit to the outside) and is the gate for all data coming into the vehicle. It supports various bus systems (Ethernet, CAN, LIN).	routing, internet	-
	Vehicle control unit	As a powertrain domain controller, the vehicle control unit (VCU) can provide torque coordination, operation and gearshift strategies, high-voltage and 48V coordination, charging control, OBD (diagnosis), monitoring, thermal management for electrified and connected powertrains in passenger cars, commercial- and off-highway vehicles. The VCU also ensures fail-operational function for highly automated driving (HAD) solutions. Other than these drive-related functions, higher-level versions also support interconnected functions like predictive and automated longitudinal guidance.	drive related functions, torque coordination, charging control, diagnosis, thermal management, fail-operational mode, automated longitudinal guidance	Conntected to Central Gateway
	Driver Assistance System Domain Controller (DASy)	The driver assistance system domain controller (DASy) is a key component with high bandwidth, computing power and memory. Meeting high security and safety requirements, it collects and merges several technologies (such as radar, video, lidar, ultrasound and highly complex functional algorithms) for a very precise 360° environment model and calculates highly complex functional algorithms for a safe and dynamic vehicle behavior - even at higher speeds.	process sensor data	Conntected to Central Gateway
	Information Domain Computer	At present, up to 15 electronic control units steer and regulate the different displays and other electronic cockpit functions in series vehicles. Prospectively, more and more functions will be merged in one central computer. The Information Domain Computer is based on a System on a Chip (SoC). The computational functions of the previously separate domains infotainment and instrumentation as well as other functions will be bundled on one processor. This saves costs, installation space, weight and consumes less energy. According to customer requirements any thinkable variant from a mere hardware solution to software integration right up to a complete solution including user interface (HMI) can be realized.	process user input, process instrument cluster data, enable entertainment functions, enable internet/wifi/bluetooth connection	Conntected to Central Gateway
	V2X Connectivity Control Unit	When vehicles are connected to one another and are able to exchange information with the infrastructure, the accident risk is reduced and pollutant emissions go down. Vehicle-to-everything communication (V2X) helps improve traffic flow and is an important step on the road to automated driving. The new capabilities of future mobility will only be able to develop their full potential with a stable and reliable data connection.	communicate with other vehicles or traffic systems, receive warnings from the cloud like traffic problems, receive security/safety over the air updates	Conntected to Central Gateway
	Body Computer Module	Body Computer Modules controls and monitors body electronics components and their functions. Advanced diagnostics make it possible to spot faults in the wiring or ancillaries and can trigger emergency programs or inform the driver of malfunctions. The BCM is an integration platform for internal and external lighting applications, access and locking functions, wipers, heating and air-conditioning functionalities as well as park assist systems and more.	access control, parking aid	Conntected to Central Gateway

Figure 5-16: Safety relevant vehicle components - Main components.

	Name	Description	Functions	Architectual Constraints
Electronic control units	Electronic Engine Control	As the main control unit, the electronic engine control unit is the heart of the engine management system. It controls fuel supply, air control, fuel injection, and ignition. Due to its scalability and enhanced performance, the control unit is also able to control the exhaust system, the transmission, and/or vehicle functions. The electronic ECU was developed for use in diesel and gasoline engines as well as for those using alternative fuels.	engine management, fuel supply control, air control, fuel injection control, ignition control, exhaust control transmission control	Conntected to VCU
	Airbag Control Unit	The airbag control unit evaluates the data from the pressure sensors to detect side crashes and from the acceleration sensors to detect side, front and rear-end crashes.	detect crashes	Connected to VCU
	ESP Unit	The electronic stability program (ESP®) supports the driver in nearly all critical driving situations. It comprises the functions of the antilock braking system (ABS) and the traction control system, but can do considerably more. It detects vehicle skidding movements, and actively counteracts them. This considerably improves driving safety.	traction control, prevent vehicle skidding	Conntected to VCU
	Electronic Immobilizer	The electronic immobilizer secures the vehicle against theft. This is achieved by a transponder with a code in the ignition key. When the ignition is switched on, that code is read by an intelligent communications interface using an antenna. If the code is valid, the electronic immobilizer releases the engine electronics system using another coded signal required for the engine to start.	secure against theft of vehicle	Connected to Body Computer Module
	Steering Control Unit	With the new modular built steering control unit (control unit with an electro motor) the EPS supports all driver assistance functions and autonomous driving. The scalable and modular design allows the highest possible flexibility when using the different steering control unit variants within a vehicle series. The steering control unit is also available as a highly integrated version and enables communication via CAN-Bus, CAN-FD or Flexray. The new steering control unit thus covers applications for all SAE levels. Functions and updates are supported by "over-the-air" technology.	support steering control, support automated driving	Connected to VCU

Figure 5-17: Safety relevant vehicle components - ECUs.

	Name	Description	Functions	Architectual Constraints
Sensor/actuator systems	Brake System	The braking system is one of the most important pieces of safety equipment in a vehicle. The components in the braking system convert the brake force applied by the driver into the required braking effect in an optimal way, ensuring the vehicle is decelerated safely and comfortably.	measure wheel speed, measure steering angle, measure acceleration	Conntected to VCU
	Electric Power Pack	The 48 V electrical power-pack P4 module is an all-in-one solution for electric drives. It integrates a 48 V electric motor, power electronics, and transmission including differential into a single compact module. It is suitable for boost and recuperation, electric all-wheel drive, and as an enabler for electric driving functions.	enable electric drive and gear automatic gear shift	Conntected to Electric Engine Controle
	Vehicle Motion and Position Sensor	For a highly precise localization, the vehicle motion and position sensor primarily makes use of satellite navigation data. In case the connection is interrupted, for example under bridges or in a tunnel, the integrated inertial sensors fill in. Yaw-rate and acceleration sensors as well as the signals of wheel-speed sensors and the steering-angle sensor precisely keep track of the vehicle's trajectory within a short time span.	localization	Conntected to DASY
	Ultrasonic Sensor	The ultrasonic sensor enables extremely comfortable parking in very small parking spaces, maneuvering in narrow situations and automatic/remote parking. The system supports emergency braking functions at low speeds through presence detection of very close objects and faster reaction to various suddenly appearing obstacles (e. g. pedestrians).	parking aid, maneuvering, automatic/remote parking, support emergency braking, obstacle detection	Connected to DASY
	Near-Range Camera	Modern vehicles often offer the driver only a limited view of the car's surroundings. Increasingly smaller side and rear windows, combined with a vehicle shape that is strongly influenced by aerodynamics and pedestrian protection, is making safe and precise maneuvering extremely difficult. The near-range camera offers the driver a better view of the vehicle's surroundings.	detect obstacles, support maneuvering, increase surroundings view	Connected to DASY
	Mid-Range Radar Sensor	The mid-range radar sensor is a bi-static multimodal radar with four independent receive channels and digital beam forming (DBF). These technologies allow the MRR to be configured with independent antennae for different directions, which improves the angular measurement accuracy and means that the radar's field of view can be adjusted depending on the situation. This technology is for example used for the side view assist.	detect obstacles in front, behind and next to the vehicle, detect braking vehicles, support distance keeping to front vehicles	Connected to DASY
	Multi-Purpose Camera	The front video camera has a key part to play in driver assistance systems because it enables vehicles to reliably detect objects and people at all times. Classic image-processing algorithms are combined with artificial intelligence methods to guarantee resilient object detection. This also makes them fit for future applications involving video-based driver assistance systems, such as automated driving.	detect obstacles in front of the car, lane departure detection, lane keeping assist, detect braking vehicles, road sign detection, construction zone detection	Conntected to DASY

Figure 5-18: Safety relevant vehicle components - Sensor and actuator systems.

No	Function	Relation type: Information, energy, substance, logic	Security threat/ safety hazard	STRIDE	Knowledge level	Recource level	Threat criticality/ severity	Security level	Security hazard impact from	Exposure	Controlability	ASIL
F1	Detect obstacle	I	ST	T	1	1	3/S3	3		E3	C3	C
F2	Forward information about detected obstacle	I										
F3	Warn driver	I										
F4	Forward warning	I										
F5	Forward warning	I										
F6	Trigger steering wheel vibration	E										
F7	React on warning	E										
F8	Possible braking	E	SH					3	F1			C
F9	Forward driver reaction	I										
F10	Forward surrounding vehicles warning	I										
F11	Forward surrounding vehicles warning	I										
F12	Forward surrounding vehicles warning	I										
F13	Warn surrounding vehicles about driver reaction	I	ST	T/D	2	2	3/S3	1	F8			

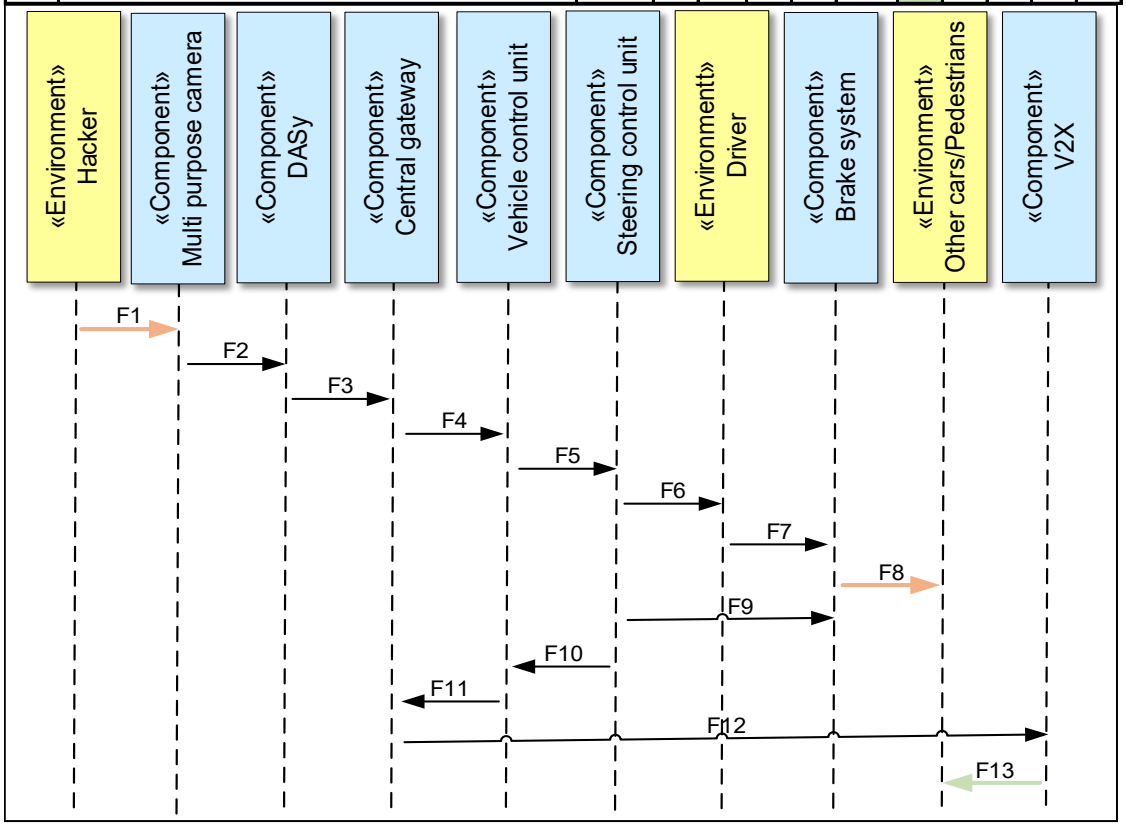


Figure 5-19: Threat identification in the white box model.

classify each function. It provides a structured and systematic way of assessing potential security threats by analysing how an attacker might exploit the system and its components. STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege.

The SAHARA resource, knowledge and threat criticality level parameters are used to identify security critical functions. The ASIL method is used to identify security critical functions. Since the SAHARA approach only describes an unclear mapping to the ASIL method, I have created a more precise mapping in Figure 5-20.

Security risk classification according to SAHARA

Threat level	Recource level											
	R0			R1			R2			R3		
	Knowledge level			Knowledge level			Knowledge level			Knowledge level		
	K0	K1	K2	K0	K1	K2	K0	K1	K2	K0	K1	K2
T0 no impact	0	0	0	0	0	0	0	0	0	0	0	0
T1 annoying, partial reduced service	3	2	1	2	1	0	1	0	0	0	0	0
T2 damage of goods, invoice manipulation, privacy	4	3	2	3	2	1	2	1	0	1	0	0
T3 life threatening	4	4	3	4	3	2	3	2	1	2	1	1

Safety risk classification according to ISO 26262 ASIL (applied if security level > 2)

Severity	Exposure	Controllability		
		C1 Simple	C2 Normal	C3 Difficult
S0	-	QM	QM	QM
S1 Light	E1 Very low	QM	QM	QM
	E2 Low	QM	QM	QM
	E3 Medium	QM	QM	A
	E4 High	QM	A	B
S2 Severe	E1 Very low	QM	QM	QM
	E2 Low	QM	QM	A
	E3 Medium	QM	A	B
	E4 High	A	B	C
S3 Fatal	E1 Very low	QM	QM	A
	E2 Low	QM	A	B
	E3 Medium	A	B	C
	E4 High	B	C	D

K0: average driver, unknown internals, K1: basic understanding of internals, K2: internals disclose, focused interests; R0: no tools, R1: standard tools, screwdriver, R2: non-standard tools, sniffer, osciloscope, R3: advanced tools, simulator, flasher

Figure 5-20: Determination of the security level and the safety level (ASIL) for the F1 function (Detect obstacle).

The sequence diagram in Figure 5-19 refines the user story "Warn Driver" (cf. Section 5-15), which has weaknesses at several points. A hacker can trigger the process by manipulating

sensor data (F1-F2), potentially causing physical damage (F7-F8). The F1 function is evaluated as follows The detection in F1 can be manipulated by tampering (STRIDE category Tampering (T)), but no direct physical damage results from this manipulation. It is therefore a potential security threat. Compared to SAHARA, no special knowledge (value: K1) and resources (value: R1) are required to detect false obstacles, so a hazardous situation can be created with little effort (value: T3/S3). This results in a high security level (level 3). For example, a teenager could place a cardboard bag on the road to trigger a reaction to detected obstacles. Due to the high criticality, the additional ASIL evaluation is performed according to SAHARA. Compared to ASIL, the exposure (value: E3) in such a case is high and the quick reaction of the driver (value: C3) is limited. This results in the high ASIL value C. Therefore, F1 and, according to this procedure, F8 (braking) must be considered as a priority in the workshop. Functions F1 and F13 have been assigned to the STRIDE tampering (T) category because both sensor data and radio traffic can be manipulated. Function F13 has also been assigned to the Denial of Service (D) STRIDE category, as the radio communication can be maliciously interrupted, e.g. by a jammer. Although F13 (warn other vehicles) appears to be critical, it does not need to be prioritised in the workshop due to its low rating.

Using the sequence diagrams modelled in the workshop, an initial system architecture can be aggregated from them (cf. Figure 5-19). The initial system architecture serves as a communication tool in the workshop for locating and discussing security-critical interfaces between individual system components. Using the ratings of each function from the sequence diagrams, critical component relationships in the initial system architecture can be highlighted for further discussion and decision making. For example, due to the high ASIL value of F8 (braking) in the sequence diagram, the relationship between the system element “brake system” and the environment element “other cars/pedestrians” in the initial system architecture is highlighted.

5.5.7 Evaluation summary and identified limitations

The approach was first tested in two internal workshops with students. In addition, the approach was tested in a one-day workshop with 30 international master students as part of the Dortmund International Summer School [**DIS20-ol**] in the Automotive (Systems) Software Engineering track. Participants were divided into 6 groups. The task of the participants was to identify use cases for autonomous vehicles. This was done by first creating a vehicle level model. Then the necessary components to realise the use cases were identified at the system architecture level and the components were related to each other. Finally, threat scenarios were identified. In total, 10 threat scenarios were identified through this approach, which I then discussed together. Through a risk analysis, 13% of the used functions could be identified as highly security critical. The one-day workshop conducted serves for the initial testing of the approach. No confident conclusions can be drawn based on it. In Section 6.2 I present a more extensive evaluation of the approach in a project with master students lasting several weeks.

5.6 Threat resolution in workshops

In this section I present the contents of a scientific paper [**JA21**]. This paper has two contributions.

I present a method that helps an interdisciplinary team of stakeholders in a workshop to solve identified threats using design patterns (C_1). The method consists of the following steps: modelling of threat scenarios in the form of SysML sequence diagrams; risk assessment of threat scenarios; derivation of a system architecture in the form of a SysML IBD; and selection and application of design patterns for security threats. The approach has been evaluated in workshops with students (C_2).

In Section 5.6.1 I explain the context and problem of this work. In Section 5.6.2 I present the literature review. In Section 5.6.3 I explain the selection and in section 5.6.4 the application of countermeasures in early system design. A summary of the evaluation and an outlook for future work is given in section 5.6.5. A detailed evaluation is described in Section 6.3.

5.6.1 Need for systematic reuse of solution knowledge

An integrated view of the system to be designed is required to address security threats early in the engineering process. This involves the involvement of multiple stakeholders from different disciplines, most of them are not familiar with security. Model-based systems engineering (MBSE) improves the understanding of systems between stakeholders through the use of models. The use of model-based design patterns enables the reuse of solutions to existing design problems in MBSE. In this work, security design patterns are used.

Based on the literature analysed (cf. Section 5.6.2), the following research question is formulated *What steps are necessary to support an interdisciplinary team of stakeholders in a workshop in such a way that they can jointly identify and resolve security threats, and how must model constructs for this purpose be designed so that they can be used in a workshop?*

In this section I will present a method for resolving threats using design patterns, which extends the method for identifying threats from Section 5.5. The method consists of the following steps Step 1: Identify security threats in the system model (cf. Section 5.5). Step 2: Select appropriate security design patterns. Step 3: Resolve the security threats in the system model using security design patterns. I will illustrate the approach using platooning as an application example (see application example in Section 5.5).

5.6.2 Analysis of related approaches

I have selected the literature according to the following criteria Number of citations, publications preferably from the last 10 years. Established security & safety approaches are partially applicable only to specific engineering disciplines, such as software engineering

[**ML06**; **MS05**; **ISO18b**; **RDG+02**; **Fer13**; **MWZ19**]. Other approaches are applicable across disciplines at the system level, but only partially address safety-relevant security threats. The SREP approach does not consider security [**RAG18**], while the following approaches do not consider security [**ADK+20**; **ISO18**; **ISO15**; **Pol16**; **Rup14**]. The Cybersecurity Guideline for Cyber-Physical Vehicle Systems [**SAE16**] and the SAHARA approach [**MSB15**] consider security risks. Unfortunately, the Cybersecurity Guideline and SAHARA do not use models. Approaches such as [**CDP+19**; **THZ17**] use models but do not use SysML. Approaches such as Security by MBRE [**Jap20**] or SAVE [**JAD21**] use SysML in the context of MBSE. However, a concrete method for resolving security issues is missing. The following approaches are suitable for use in workshops and support the identification of new security vulnerabilities: [**JKK20**] supports stakeholders in visualising threat cases using a 3D environment, while [**TKA+19**] extends design thinking to consider security. However, both approaches do not support the resolution of identified vulnerabilities. Furthermore, I analysed several sources in which security or safety design patterns were presented [**PADK+20**; **THZ17**; **CDP+19**; **Fer13**; **MWZ19**]. Most of the design patterns were unsuitable in terms of structure, description and presentation for use in workshops with an interdisciplinary team of stakeholders for the following reasons: Most of the design patterns were intended for application by IT experts. This was reflected in the very high level of detail for IT systems and also in the use of UML instead of SysML. On the other hand, design patterns tailored for use in MBSE did not consider the security aspect.

5.6.3 Selecting appropriate security design patterns

Based on the identified threats, appropriate design patterns are selected from a catalogue and applied. A design pattern must be described by attributes such as name, application context, problem description, etc. Such attributes support the selection of appropriate design patterns. In order not to hinder the creativity process in the workshop, a quick selection of solution patterns must be ensured by fulfilling the following requirements: *R1: Design patterns must be described only by the most important attributes, while the text must not contain unnecessary details. R2: The idea of the design pattern must be understood immediately. No complicated model constructs must be used. Easy-to-understand examples with easy-to-understand model elements support this. R3: Different model elements need to be identified for quick visual recognition. Color schemes in conjunction with stereotypes provide support here.*

I have analysed the following security pattern catalogues, which do not fully meet the above requirements: [**ADK+20**; **THZ17**; **CDP19**; **Fer13**]. In particular, none of these approaches satisfy R2 and R3. In order to satisfy R1-R3, I propose the following template, which I illustrate using the DIDS design pattern. The idea of the Distributed Intrusion Detection System (DIDS) design pattern is to use an attack database that the system accesses (cf. Figure 5-21).

ID: 0007	Name: (Distributed) Intrusion Detection System
Summary An intrusion detection system (IDS) is a system for detecting attacks in the vehicle system. The IDS can extend existing security mechanisms such as the firewall and thus increase the security of the vehicle system. Potential attacks on the vehicle system are compared with attacks from a database and an alarm is triggered if they match. With a distributed IDS (DIDS), several IDS are integrated in the vehicle system. In this case, attacks occurring in and between several vehicle components are detected via several IDSs.	
Context IDS can be integrated into any vehicle component that processes data. At least 2 IDS are required for a DIDS, which must also be able to communicate with each other. Application areas are: System of Systems architecture (Car2X), system architecture (in-vehicle networks) and component architecture (ECU internal, communication interfaces).	**Constraints and Consequences** Only those attacks can be detected which correspond to the attacks from the database or which are similar to them. Unlike the Intrusion Prevention System, the IDS does not prevent attacks. The IDS/DIDS can itself serve as a target for attacks.
Relationship with Other Patterns Compatible with 0003 Defense in Depth Design Pattern	**Problem** Intrusion detection is the problem of detecting attacks on a vehicle system or on a vehicle component. In contrast to simple IDS, DIDS can detect attacks on and between several vehicle components.
Solution The IDS checks and processes the collected data during pattern recognition and compares it with identical or similar signatures from the pattern database. If events apply to one of the patterns, an alarm is triggered. In a DIDS, signatures from several IDSs are stored and compared.	
Example Road markings can be dirty or have stickers. If a group 1 vehicle reacts incorrectly to such lane markings and the driver takes appropriate action, this data is sent to the fleet backend. In this way, other vehicles (group 2) can be warned.	

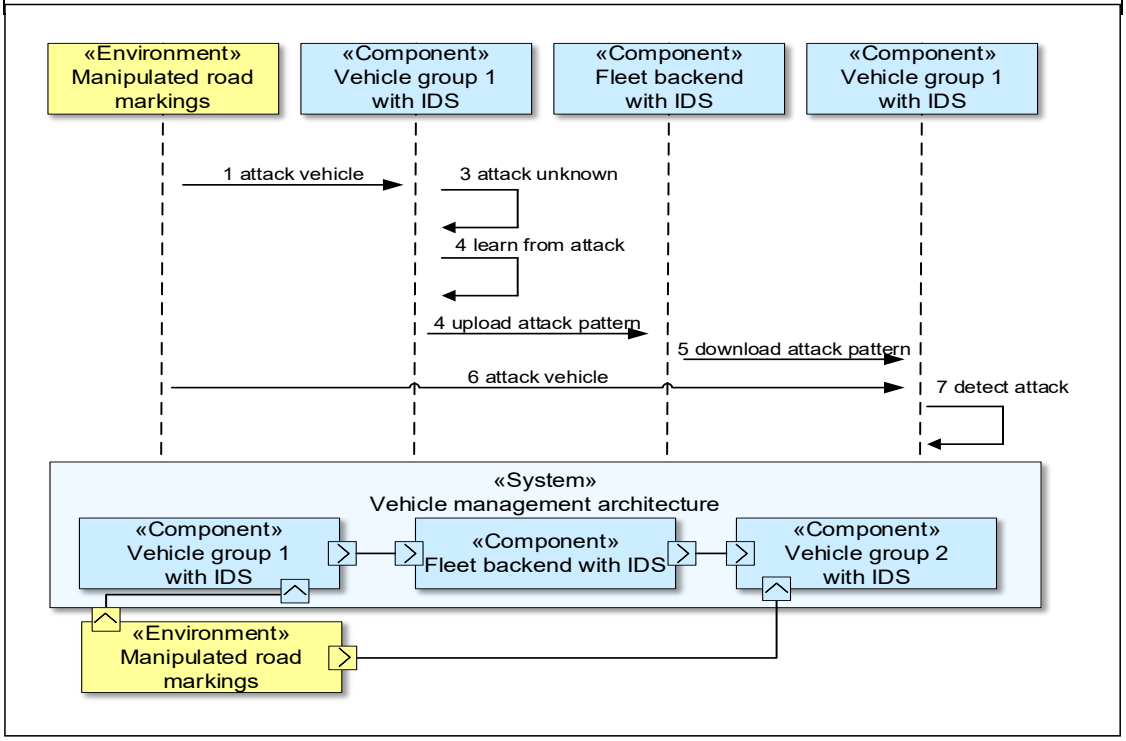


Figure 5-21: (Distributed) Intrusion Detection System Design Pattern

5.6.4 Threat resolution using design patterns

In this section I present how the security threats identified in the workshop can be addressed using design patterns. This is based on a system model. I illustrate the resolution of security threats using the DIDS security design pattern (cf. Figure 5-21). This results in a redesign of the system model in terms of structure and behaviour (cf. Figure 5-22).

The DIDS design pattern uses a local database that synchronises with a central database. To resolve security threats in the system model, sensor data from the *Multi purpose camera* must be compared with data from the attack database (cf. Figure 5-22, F3-F4). For example, if the platoon leader (local attack database) or the vehicles of the fleet (central attack database) have detected previously placed cardboard bags as uncritical obstacles, a false alarm can be avoided (F8 & F14).

Based on the analysis of the design pattern, elements of the system architecture relevant for redesign are identified. The *Central gateway* component is relevant because all data converge in this component. It is therefore appropriate to extend this component to include an attack database. This allows incoming data to be directly compared with attacks from the database (F3-F4). For synchronisation of identified attacks between different vehicles, the *Fleet backend IDS* is added to the system architecture as an environment object (cf. Figure 5-22). The *Central gateways* of the individual vehicles synchronise with the *Fleet backend* regarding identified attacks (F1 & F15).

The sequence diagram of the DIDS design pattern is used to redesign the sequence diagrams of the system model. After applying the DIDS pattern, the following functional sequence results (cf. Figure 5-22): The local attack database is synchronised at regular intervals (F1). If an obstacle is detected, the data is compared with the local attack database (F3 & F4). If it is negative, the driver is warned by a vibration in the steering wheel that an obstacle has been detected on the road (F5 & F8). If the driver reacts, e.g. by braking (F9), the information is communicated to the surrounding vehicles (platoon members) (F14). Based on the driver's reaction, the detected object is assessed as dangerous or not, and synchronised with the central attack database. By applying the design pattern, new components, functions and relationships were added to the system model.

A subsequent re-evaluation (see the sequence diagram in Figure 5-22) shows that the application of the DIDS design pattern has had an impact on the security and safety level. In this case, the security level of the functions has been reduced (see functions F1, F3, F4 & F10). In addition, new security threats have appeared with F14 and F15, which have been rated as non-critical. Overall, the ratings serve as a decision support for stakeholders. If the rating decreases after applying a design pattern, the application was successful. If the rating increases, the stakeholders must decide whether the application of the design pattern should be discarded or whether another design pattern should be applied to solve the problem.

No	Function	Relation type: Information, energy, substance, logic	Security threat/ safety hazard	STRIDE	Knowledge level	Recource level	Threat criticality/ severity	Security level	Security threat impact from	Exposure	Controlability	ASIL
F1	Distribute attack database update	I	ST	T	2	2	3/S3	1				
F2	Forward update	I										
F3 & F4	Detect obstacle and check local attack database	I	ST	T	2	2	3/S3	1				
F5	Warn driver	I										
F6	Forward warning	I										
F7	Forward warning	I										
F8	Trigger steering wheel vibration	E										
F9	React on warning	E										
F10	Possible braking	E	SH					1	F1, F3			
F11	Communicate driver reaction	I										
F12	Forward driver reaction	I										
F13	Forward driver reaction	I										
F14	Warn surrounding vehicles	I	ST	T,D	2	2	2/S3	0	F10			
F15	Upload driver reaction to backend attack database	I	ST	T,D	2	2	2/S2	0	F10			

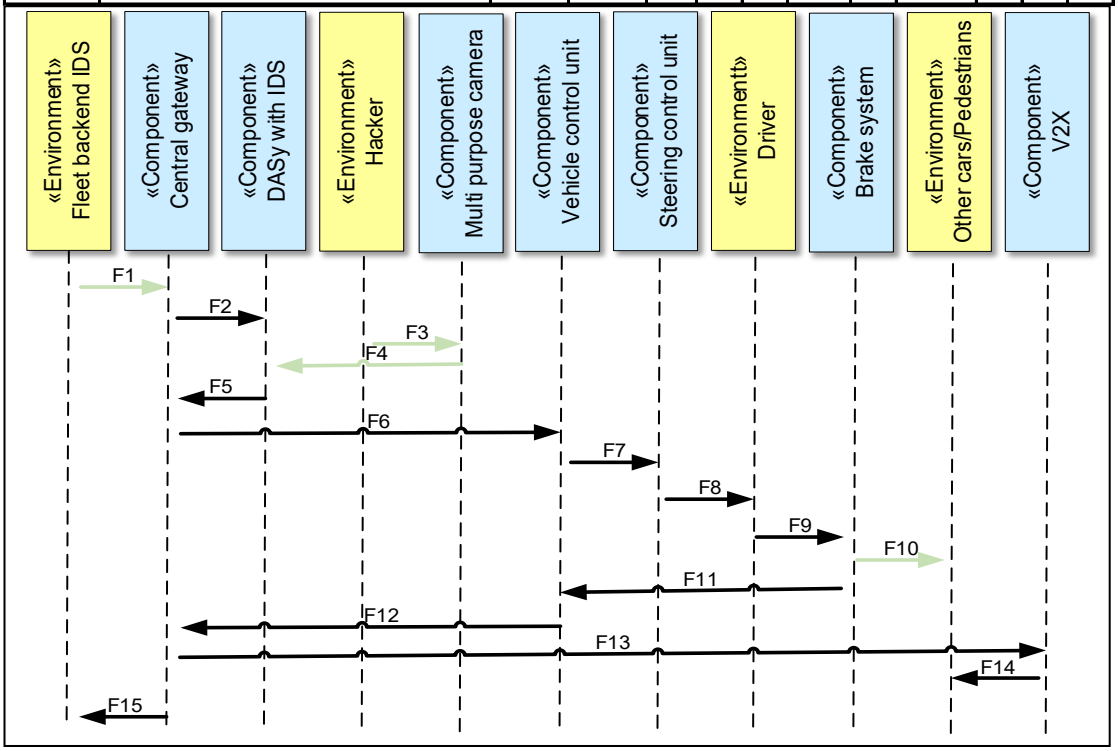


Figure 5-22: Resolving security threats in the initial system model by applying the DIDS design pattern

5.6.5 Evaluation summary and identified limitations

As part of a project at the University of Paderborn, I tested the approach with 67 master students from the fields of computer science, computer engineering and business informatics. The project was carried out using platooning as an example and lasted 8 weeks. A total of 21 groups of about three people used the method to identify and address security threats in the system design phase. The goal of the evaluation was to test the applicability of the method in workshops and to identify potential improvements for future work. In particular, I wanted to find out which steps/tools/designs needed to be adapted to increase the effort/benefit ratio when applying the method. Overall, all teams were able to identify security threats in joint workshops and solve most of them using the design patterns provided (cf. Section 6.3 for a comprehensive evaluation of the approach). Unfortunately the students were only able to resolve a limited number of threats because there were not enough Security Design Patterns (SDPs) available. This was improved in a project the following year (cf. Section 5.7) by providing more SDPs.

5.7 Security design patterns

In this section I present the contents of a scientific paper [**JFA+23**]. This paper has two contributions: 10 Security Design Patterns (SDPs) have been created for use in early system design (C_1). The SDPs were originally created as part of a master's thesis that I supervised [**Fah21**]. The application of the SDPs was evaluated as part of an 11-week project (cf. [**UPB21**]) with 140 master students (C_2).

5.7.1 The need to use security design patterns in early system design

ISO/SAE 21434 [**ISO21**] requires the createion of a system architecture and defines requirements for a comprehensive risk analysis to be carried out at the concept phase. Based on the risks identified in the system architecture, countermeasures shall be identified in the concept phase. UNECE R155 [**UN22**] lists 24 countermeasures against cyber attacks. The result of the concept phase is an initial vehicle system architecture. The work in the concept phase is characterised by the collaboration of several experts in workshops [**Jap21**]. To implement countermeasures in the initial system architecture, the textual listing of countermeasures from UNECE R155 is not sufficient. More comprehensive countermeasure information describing the problem to be solved is required. Based on my project experience, models of countermeasures support the redesign of the initial system architecture. Suitable tools are Security Design Patterns (SDP). To enable an interdisciplinary team to apply such design patterns in workshops, the descriptions must be generally understandable and the models must consist of simple elements of a modelling language. According to the literature review (cf. Section 5.7.2), there are no suitable approaches or sources of design patterns for the automotive domain that can be applied in early system design by an interdisciplinary team of experts. [**JFA+23**] addresses the research question: How must SDPs be defined so that they can be used during development

by an interdisciplinary team to define security countermeasures?

In Section 5.7.2 I present the literature review. In the appendix in Section A.2, I list the 10 SDPs that have been developed, of which I am not the main author. In Section 5.7.3 I present the summary of the evaluation. The detailed evaluation is presented in Section 6.4.

5.7.2 Analysis of related approaches

In this section, I present the analysis of the papers related to SDPs (cf. Figure 5-23). I evaluate the approaches on the basis of a literature review according to four requirements. *R1: The design patterns have been described using easy to understand text.* The early system design phase is characterised by collaboration between stakeholders from different disciplines. These stakeholders are often unfamiliar with the technical details of other disciplines. *R2: The design patterns use simple constructs of a modeling language.* This is necessary because in early system design not all stakeholders are experts in modelling system architectures. *R3: The design patterns need to include solutions for security threat resolution.* This provides alignment with UNECE R155 and ISO/SAE 21434. *R4: The design patterns must support the prevention of safety hazards.* This provides alignment with ISO 26262 [**ISO18**]. Security in the vehicle is always about ensuring safety. *R5: The considered approach must contain several design patterns, e.g. in the form of an initial catalog.*

R1	The design patterns have been described using easy-to-understand text.
R2	The design patterns use simple constructs of a modeling language.
R3	The design patterns need to include solutions for security threat resolution.
R4	The design patterns must support the prevention of safety hazards.
R5	The considered approach must contain several design patterns.

To what extent do the appraoches considered satisfy the requirements?

Satisfied | Partially s. | Not satisfied

	Considered approaches	**Requirements** R1	R2	R3	R4	R5
[ADK+20]	Pattern-based systems engineering	■	■			1
[*JA21]	Resolution of security threats in the system architecture	■	■	■	■	1
[THZ17]	Systematic pattern approach	■		■	■	1
[MMS+20]	Safety & security pattern engineering approach	■		■	■	2
[CDP+19]	Security patterns for automotive systems	■		■	■	10
[CDP+20]	Security patterns for connected automotive systems	■		■	■	10
[Fer13]	Security patterns in practice	■		■	■	80

Figure 5-23: Rating of existing work.

I illustrate the rating with the following approaches: The [**JA21**] approach is designed for implementation in the concept phase and uses a textual description that does not use deep technical details (R1 fulfilled). The approach uses simple model constructs (R2 satisfied) and primarily considers safety (R3 satisfied) with an impact on safety (R4 partially satisfied). The approach presents only one design pattern (R5 not satisfied). The [**Fer13**] approach describes a method for using SDPs and provides an extensive catalogue of 80 patterns (R3 and R5 fulfilled). The design patterns contain too much technical detail

for the concept phase (R1 partially fulfilled) and use complex model constructs that can only be understood by modelling experts (R2 not fulfilled). The patterns focus on IT systems and are rarely related to safety (R4 partially satisfied). The [**CDP+20**] approach describes 10 design patterns for the automotive sector (R5 fulfilled). The design patterns represent countermeasures against security threats (R3 fulfilled) and are safety relevant (R4 fulfilled). Unfortunately, the descriptions require a deep understanding of security (R1 not fulfilled). Furthermore, the complex and detailed UML models can only be understood by modelling experts (R2 not fulfilled). In particular, the design patterns cannot be used in workshops with domain experts from different disciplines for early system design.

5.7.3 Evaluation summary and identified limitations

As part of an 11-week project with Masters students, an initial vehicle system architecture was created and then a risk analysis was carried out. Countermeasures were selected and applied to reduce the risk. 28 teams were formed out of 140 students. 18 teams decided to use SDPs as countermeasures in Variant A. 10 teams in Variant B decided to use countermeasures from publicly available scientific papers. I quantitatively evaluated the results of all 28 teams on several indicators. I did not find any significant quantitative differences. The number of diagrams and functions, the security and safety ratings and the resulting risk of both variants were almost the same. I also looked at the feedback from all the teams. The following points stood out. The teams in Variant B often mentioned the high effort required to find suitable countermeasures (6 out of 10 teams). Furthermore, in Variant B, 9 out of 10 teams stated that it was very time-consuming to understand and apply the countermeasures in the scientific publications they found. I suspect that the Variant B approach is not attractive enough for repeated use of risk analysis when applying multiple countermeasures in succession because of the increased time required. SDPs are a better choice in this context (cf. Section 6.4 for a comprehensive evaluation of the approach). Further indicators are missing to determine the differences between the two variants more precisely. In addition, the approach needs to be conducted with experts from the automotive industry to obtain further suggestions for improving the description and design of the SDPs.

5.8 Derivation of requirements from models

In this section I present the contents of a scientific paper [**JAK+21**]. This paper has two contributions. The approach supports the derivation of initial security requirements from models (C_1). The approach has been evaluated in a project with students (C_2).

In Section 5.8.1 I explain the context and problem of the paper. In Section 5.8.2 I present the literature review. In Section 5.8.3 I explain the derivation of security requirements from black-box models, and from white-box models in Section 5.8.4. A summary of the evaluation and an outlook for future work is given in Section 5.8.5. A detailed evaluation is described in Section 6.3.

5.8.1 Need to derive security requirements from models

A holistic view of the system to be designed is necessary to identify and resolve security threats in engineering at an early stage. This includes the involvement of multiple stakeholders from different disciplines, most of them are not familiar with security.

Model-based systems engineering (MBSE) improves the understanding of the system between stakeholders through the use of models. Conducting workshops in the context of MBSE promotes interaction between stakeholders, so that confusion between stakeholders, especially regarding security, can be identified and resolved already in the workshop. Due to the increasing digitalisation, online workshops are possible and due to the current pandemic (COVID-19) also necessary. Based on the created models, requirements can be derived and discussed in the workshop.

I formulate the following research questions RQ_1*: How can MBSE be used so that an initial system model can be created in workshops by a team of stakeholders considering security & safety?* RQ_2*: How can models be used to derive (safety-related) security requirements?* RQ_1 is already solved by the SAVE approach (cf. Section 5.5). Since none of the approaches analysed in Section 5.8.2 satisfy RQ_2 in conjunction with RQ_1, I will present in Sections 5.8.3 and 5.8.4 how to derive security requirements from SAVE based on the models created.

I will present a method consisting of the following steps: Step 1: Identification of security threats from a black box perspective. Step 2: Derive black-box security requirements. Step 3: Creation of a white-box system model. Step 4: Derive white-box security requirements.

Steps 1 and 3 have already been presented in the Sections 5.5.4 & 5.5.6 and illustrated with the example of platooning. In the following sections I will explain how to carry out Steps 2 and 4. I extend the platooning example from Section 5.5.3.

5.8.2 Analysis of related approaches

The approaches [**ML06**; **MS05**; **ISO18b**; **RDG+02**; **Fer13**; **MWZ19**] specifically consider the security aspect of RQ_1. However, these are designed for use in the context of software-intensive systems development. The approaches [**ADK+20**; **Pol16**; **Rup14**; **ISO18**; **ISO15**; **CDP+19**; **THZ17**; **Jap20**; **RAG18**; **SAE16**] are designed for use in the context of developing intelligent mechatronic systems. However, these approaches are too broad for use in workshops without extensive customisation, so RQ_1 is only partially addressed. The SCIL approach [**WJK+20**] supports requirements elicitation (RQ_2), unfortunately the approach does not use models (RQ_1) and does not consider safety & security. The CONSENS approach [**GRS14**] supports the creation of models (RQ_1) in workshops and supports the derivation of requirements (RQ_2). Unfortunately, the approach does not consider safety & security. The CONSENS 3D approach [**JKK20**] uses a 3D environment to visualise security & safety related use cases and allows the derivation of SysML models (RQ_1). Unfortunately, the approach does not support SRSR derivation (RQ_2). The

RE-EDIT approach [**JKK20**] uses security & safety design patterns to resolve threats in models (RQ_1). Unfortunately, the approach does not consider the derivation of requirements (RQ_2). The SAHARA approach [**MSB15**] supports risk assessment. Unfortunately it does not use models (RQ_1).

5.8.3 Derivation of security black box requirements

The goal of this step is to derive black box security requirements (cf. Figure 5-24) based on the black box model from Step 1 presented in Section 5.5.4.

The black box architecture and sequence diagrams are used in the workshop to identify and map the structural relationships and processes between the system to be developed and its environment. The requirements specify and supplement the short information of the model elements with free text information. In the example, the requirement *(B-R01) The Platoon Leader Vehicle (PLV) must check if a detected obstacle is real and dangerous.* extends the description of the modelled relationship *Manipulated sensor data* in the architecture model between the model elements *Vehicle* and *Hacker*.

In the workshop, requirements are created in sub-groups for the purpose of parallelization. In addition, requirements are categorised as security or safety relevant. The categorisation is based on the stereotypes defined in the SysML models. It should be noted that there is no 1:1 relationship here. Several security/safety requirements can describe a model element, and several model elements can be described by a comprehensive security/safety requirement. In the case of the *B-R01* requirement, the processing of potentially manipulated sensor data is security relevant but not directly safety critical.

ID	Requirement description	Safety/ Security relevant	Derived from
B-R01	The platoon leader vehicle (PLV) must check if a detected obstacle is real and dangerous.	Security	User story "Warn driver"; IBD "Vehicle"; Sequence diagram "Detect obstacle"
B-R02	When obstacles are detected, the PLV must not cause any damage to its surroundings by its reaction.	Safety	User story "Warn driver"; IBD "Vehicle"; Sequence diagram "Detect obstacle"
B-R03	If a detected obstacle is classified as real and dangerous, the driver of the PLV must be warned by a vibration on the steering wheel. Furthermore, platoon follower vehicles (PFV) must be warned about the detected obstacle.	Both	B-R01
B-R04	If a detected obstacle is not classified as real and dangerous, the driver of the PLV must not be warned by a vibration on the steering wheel.	Safety	B-R03
B-R05	If an obstacle detected by the PLV is classified as not real and dangerous, the PFVs must not be warned.	Security	B-R03

Figure 5-24: Illustration of the artefacts created in Phase 1 (Black Box Model) & Phase 2 (Black Box Requirements).

5.8.4 Derivation of security white box requirements

The goal of this step is to derive security requirements based on the white box model presented in Section 5.5.6. The created requirements (cf. Figure 5-25) are the end result of the approach.

The white box architecture and sequence diagrams contain information about the identified critical components and component functions. In addition, each function is described by a number of attributes. This information is used to derive security requirements.

Each requirement has a description, a categorisation in terms of security/safety relevance, a security level and an ASIL. In addition, I indicate whether a requirement is a refinement of a black box requirement or whether a requirement is new. To establish traceability of requirements, the associated white box models and the underlying black box requirements are linked.

By analysing the white box system architecture, and in particular functions F1 - F8 of the white box sequence diagram, the following requirement is derived *(W-R01) When obstacles are detected, the driver of the platoon leader vehicle (PLV) must be warned by a vibration on the steering wheel, taking into account the referenced function sequence.*

As functions F1-F8 are safety related, requirement W-R01 is also set as safety related. The other attribute values are copied in the same way. If a requirement addresses multiple functions with different ratings, the maximum values are used. The requirement *W-R01* is not a refinement of a requirement from Step 2 and is therefore classified as new.

To ensure the traceability of requirements, they must be linked to the associated model elements from Figure 5-22. When requirements are refined, this must also be specified for traceability purposes.

Unlike Requirement *W-R01*, Requirement *(W-R03) The Advanced Driver Assistance System must check whether an obstacle detected by the radar sensor is real and dangerous* is a refinement of Requirement *B-R01* from Step 2. The requirement has been extended to include the naming of specific components. Instead of *Platoon Leader Vehicle (PLV)*, the specific component *Radar sensor* required for object detection is now used. In addition, the *Advanced Driver Assistance System* is now used instead of *PLV* to classify a detected object as safety relevant. This information was not available after step 2.

Non-compliance with the *W-R03* requirement can be caused by sensor manipulation, for example, so that object detection fails. This is a security vulnerability. However, it is not safety critical as failure of the vehicle to comply with this requirement cannot cause direct physical damage.

ID	Requirement description	Safety/ Security relevant	Security/ Safety level		Refined/ New req.	Derived from
W-R01	When obstacles are detected, the driver of the platoon leader vehicle (PLV) must be warned by a vibration on the steering wheel considering the referenced function sequence.	Both	3	C	New	IBD "Vehicle"; Sequence diagram "Detect obstacle" [F1-F8]
W-R02	If the PLV initiates emergency braking or evading, the platoon follower vehicles (PFVs) must be warned of it considering the referenced function sequence.	Security	1		New	W-R01; Sequence diagram "Detect obstacle" [F8-F13]
W-R03	The advanced driver assistance system must check, if an obstacle which was detected by the radar sensor is real and dangerous.	Security	3		Refined	W-R01; B-R01
W-R04	When obstacles are detected by the radar sensor, the PLV must not cause any damage to its surroundings like other cars or pedestrians by its reaction. The reaction must base on the decision of the advanced driver assistance system or the driver of the PLV.	Safety		C	Refined	W-R01; B-R02
W-R05	If a detected obstacle is classified by the advanced driver assistance system as real and dangerous, the driver of the PLV must be warned by a vibration on the steering wheel. Furthermore, PFV must be warned about the detected obstacle by the vehicle2X communication unit.	Both	3	C	Refined	W-R02; B-R03
W-R06	If a detected obstacle is not classified by the advanced driver assistance system as real and dangerous, the driver of the PLV must not be warned by a vibration on the steering wheel.	Both	3	C	Refined	W-R01; B-R04
W-R07	If an obstacle detected by the radar sensor of the PLV is classified by the advanced driver assistance system as not real and dangerous, the PFVs must not be warned by the vehicle2X communication unit.	Security	1		Refined	W-R02; B-R05
W-R08	A PFV must check that the warning message comes from the PLV and that it has not been tampered with by an attacker during transmission.	Security	1		New	W-R07; Sequence diagram "Detect obstacle" [F13]
W-R09	If an attacker performs a DOS attack on a PFV in the form of a message flood, the driver of the PFV must be informed that the platoon communication service is temporarily unavailable.	Security	1		New	W-R07; Sequence diagram "Detect obstacle" [F13]

Figure 5-25: Illustration of the artefacts created in Phase 2 (White Box Model) & Phase 4 (White Box Requirements).

5.8.5 Evaluation summary and identified limitations

The approach was carried out as part of an 8-week project with 67 students from Computer Science, Computer Engineering and Business Informatics. The students were divided into 21 teams. First, the students had to create a vehicle level model (black box model) and derive black box security requirements. Then they had to identify the necessary components and component relationships for the system architecture (white-box model). Based on this, white-box security requirements had to be derived. In general, all teams were able to derive security requirements from the models using the approach. On median, 55% of the security requirements for all teams could be refined by the additional creation of the white-box model (cf. Section 6.3 for the comprehensive evaluation of the approach). The results of the evaluation with the students are an indicator that the use of models improves the common understanding of the system to be developed and thus increases the quality of the requirements. Whether this finding can be confirmed with subject matter experts from the automotive industry remains to be verified in the future.

5.9 Procedure model for the development of a cybersecurity concept

In this section, I present the general procedure of my approach (cf. Figure 5-26). I have split the creation of the 15 ISO/SAE 21434 work products for the concept phase into 5 phases. In addition, I consider the system design in the concept phase on two levels. At the first level, I consider the system to be developed as a black box. Here, the focus is on the interfaces of the system to interacting environmental systems or environmental elements. On the second level, I consider the system to be developed as a white box. Here, the focus is on the system components and their relationships to each other. This avoids mixing levels of abstraction. For example, discussions of specific details are moved to later phases, while the big picture can be considered first. In the first three phases, the analysis is carried out at the black-box level, and in the last two phases at the white-box level. Figure 5-26 shows in which phase, which work products are created.

A SysML-profile enables the adaptation of SysML to specific application purposes. For the implementation of all 5 phases, a SysML-profile created by me is used (cf. Section 6.6.8). The SysML-profile contains specific model constructs, attributes, inheritance relationships, and model relationships adapted to ISO/SAE 21434, which facilitate the creation of the work products of the concept phase. The SysML-profile, including a continuous application example, is the implementation of my work in a professional MBSE tool (cf. Section 6.6).

In the following, each phase is described in detail.

		Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Concept phase	[WP-09-01] Item definition	x			x	
	[WP-09-02] TARA		x	x		x
TARA methods	[WP-15-01] Damage scenarios		x			
	[WP-15-02] Assets with cybersecurity properties		x			
	[WP-15-03] Threat scenarios			x		
	[WP-15-04] Impact ratings with associated impact categories		x			
	[WP-15-05] Attack paths					x
	[WP-15-06] Attack feasibility ratings					x
	[WP-15-07] Risk values					x
	[WP-15-08] Risk treatment decisions					x
	[WP-09-03] Cybersecurity goals					x
	[WP-09-04] Cybersecurity claims					x
	[WP-09-05] Verification report for cybersecurity goals					x
	[WP-09-06] Cybersecurity concept					x
	[WP-09-07] Verification report of cybersecurity concept					x

Figure 5-26: Procedure model for the creation of the 15 work products of the concept phase of ISO/SAE 21434.

5.9.1 Phase 1: System analysis at environment level

[WP-09-01] Item definition

This phase deals with the work product [WP-09-01], the definition of the item. According to ISO/SAE 21434, an item is a component or a set of components that realizes a function at vehicle level. I split the definition of the item into two phases. In Phase 1, the item is considered as a black box. Here, the focus is on identifying the elements that interact with the item. The consideration of the item as a white box is done in Phase 4 (cf. Section 5.9.4). This prevents mixing information from different levels of abstraction. The starting point is the consideration of use cases. In my approach, use cases are functions on the vehicle level, e.g. traffic sign recognition. The following tool is used to collect use cases:

- SysML use case diagrams (cf. Section 2.2.5.1)

SysML use case diagrams allow the modeling of the item as a black box. Several use cases can be assigned to the item. At the same time, actors interacting with the item can be identified (e.g. traffic signs). Furthermore, it can be determined whether use cases assigned to the item actually belong to the item or have to be realized by another system. By considering the item as a black box, the focus can be placed on the interaction of the item with its environment. This forms the basis for the definition of interfaces in Phase 4. The use case diagram is a starting point for deriving black box requirements. The following tool is used to document requirements:

- SysML requirements in table form (cf. Section 2.2.5.6)

When defining requirements, ambiguities in the modeling can be identified. This offers the potential to improve the quality of the model. At the same time, these requirements can serve as documentation for people who did not participate in Phase 1, such as members of

other departments, affiliated companies, or suppliers. The use of SysML use case diagrams and SysML requirements enables traceability to be ensured. For this purpose, requirements are linked to the use cases, the item, and the interacting actors.

Result of Phase 1

The result of Phase 1 is the item definition [WP-09-01] in the form of a use case diagram, which considers the item as a black box. For a better understanding of the item definition by external parties, the black box requirements serve as support.

5.9.2 Phase 2: Impact analysis at environment level

[WP-15-01] Damage scenarios

ISO/SAE 21434 requires the identification of damage scenarios [WP-15-01]. A damage scenario is an adverse consequence on a vehicle or a vehicle function with consequences for a road user, e.g. the deception of a vehicle's traffic sign recognition system resulting in an abrupt braking with collision. The following tool can be used to identify damage scenarios:

- 3D environment for identifying damage scenarios (cf. Section 5.2)

With the help of the 3D environment, damage scenarios can be modeled and discussed. Through visualization, ambiguities between stakeholders can be clarified already in the concept phase. The following tool is used to document damage scenarios:

- SysML use case diagrams (cf. Section 2.2.5.1)

With the help of the use case diagrams, the damage scenarios are collected and linked to the use cases from Phase 1. Damage scenarios are distinguished from normal use cases in use case diagrams with the help of a separate SysML stereotype. This makes it clear which damage scenarios are associated with which use cases or which damage scenarios can result from which use cases. The following tool is used for the detailed description of the damage scenarios:

- SysML requirement tables (cf. Section 2.2.5.6).

The damage scenarios modelled in the 3D environment can be exported as images and linked to the descriptions of the damage scenarios in the SysML Requirements tables. This improves the understanding of the damage scenario descriptions during a later review, especially for people who were not involved in Phase 2.

[WP-15-02] Assets with cybersecurity properties

In [WP-15-02], ISO/SAE 21434 requires the identification of assets and their cybersecurity properties whose compromise leads to a damage scenario. An asset is an object (e.g. a component) that has value or contributes to value. An asset has one or more cybersecurity properties, the compromise of which may lead to one or more damage scenarios. A cybersecurity property is an attribute that has to be protected. Such properties are confidentiality, integrity, and/or availability according to ISO/SAE 21434. For each damage scenario, it is investigated which asset (e.g. multi-purpose camera) is affected and which cybersecurity property has been compromised (e.g. integrity).

[WP-15-04] Impact ratings with associated impact categories

In the context of [WP-15-04], damage scenarios must be assessed in terms of their potential negative impact on road users in the categories of safety, financial, operational, and privacy (S, F, O, P). With regard to safety, reference is made to the ASIL classification scheme of ISO 26262 (cf. Section 2.3.1). Here, the additional parameters Exposure and Controllability are used together with a mapping table to determine the safety level. The following tool is used to evaluate damage scenarios from a safety perspective:

- Statistically based ASIL Tables (cf. Section 5.3)

The tables are based on data from the Federal Statistical Office with 10 million police-registered road traffic accidents in Germany. I aggregated this data and combined it with the ASIL risk classification scheme. Thus, this data can be used in workshops of the concept phase.

Result of Phase 2

The result of Phase 2 is documented damage scenarios [WP-15-01], which are supplemented by a 3D visualization. For these damage scenarios, affected assets and compromised cybersecurity properties are identified [WP-15-02]. Based on this, an impact analysis and assessment is carried out, partly using statistical data [WP-15-04].

5.9.3 Phase 3: Security analysis at environment level

[WP-15-03] Threat scenarios

ISO/SAE 21434 requires the identification of threat scenarios [WP-15-03]. A threat scenario is a possible cause for compromising the cybersecurity properties of one or more assets in order to realize a damage scenario. An example of a threat scenario is the manipulation of a vehicle's traffic sign recognition by displaying a traffic sign on a digital signage system. This may cause the vehicle to abruptly reduce its speed, resulting in a collision with a closely following vehicle.

According to ISO/SAE 21434, the identification of threat scenarios is based on the damage scenarios already documented in Phase 2. Threat scenarios are searched for that lead to the damage scenarios already identified. In contrast to the reverse procedure, this avoids potentially investing a lot of effort in identifying threat scenarios without significant impact. The following tools are used to collect and document the threat scenarios: [WP-15-03].

- SysML use case diagrams (cf. Section 2.2.5.1)
- SysML requirements in table form (cf. Section 2.2.5.6)

SysML use case diagrams are used to collect threat scenarios. Threat scenarios are distinguished from normal use cases in use case diagrams with the help of a separate SysML stereotype. Here, it can be identified which external actors (e.g. other systems, an attacker) cause the threat scenario. SysML requirements are used to describe the threat scenarios in detail. Documenting the threat scenarios improves the understanding of persons who were not involved in Phase 3 but have to work with this information in subsequent phases. A link is established between the threat scenarios from Phase 3 and the damage scenarios from Phase 2 to ensure traceability. I use the following tool in order to establish the reference to UN R155:

- UN R155 threats (cf. Section 2.1.1)

In order for a vehicle to be approved to UN R155, it must be shown that the 77 threats of UN R155 have been considered. In my approach, I integrate the consideration of the UN R155 threats into this phase. This involves, on the one hand, examining which already identified threat scenarios correspond to which UN R155 threats. And on the other hand, it is examined which UN R155 threats are relevant to the already identified damage scenarios from Phase 2. This then forms the basis for the textual description of the threat scenarios.

Result of Phase 3

The result of Phase 3 are threat scenarios [WP-15-03], which are documented in the form of use cases and requirements and linked to the damage scenarios from Phase 2. To ensure vehicle approval according to UN R155, the threat scenarios are compared with the threats of UN R155, or threat scenarios are derived from UN R155.

5.9.4 Phase 4: Analysis at system level

[WP-09-01] Item definition at system level

According to ISO/SAE 21434, an item definition [WP-09-01] shall be created. An item is a component or a set of components that realizes a function at vehicle level. In Phase 1, the item was already considered from a black box perspective. In this phase, the item is considered from a white box perspective. For this purpose, the necessary components and component relationships are identified and modeled for the use cases to be realized

from Phase 1. This phase incorporates my results from a 3-year industrial project with a German premium car manufacturer. The following tools are used to create the item:

- Effect Chain Modelling Language (ECML) (cf. Section 5.4.2).
- Mapping between ECML and SysML (cf. Section 5.4.2)
- SysML Internal Block Diagram (IBD) (cf. Section 2.2.5.3)
- Safety relevant vehicle components (cf. Section 5.5.4)
- Prototype for model transformation from ECML to SysML models (cf. Section 5.4.5)

Based on my project experience, I make the following assumptions: The work in the concept phase takes place in an interdisciplinary team in workshops. The participants come from different fields such as mechanical engineering, electrical engineering, physics, computer science, and controlling. This means that only simple modeling constructs can be used as a "common denominator". In addition, the participants are often managers who have holistic knowledge in several development areas. Since managers often have little time, the focus in such workshops has to be on content considerations and not on observing exact modeling rules. Formalization of the models can be done later by a modeling expert. Since the use of professional MBSE (Model-Based Systems Engineering) tools requires training and the creation of models requires simultaneous attention to modeling syntax, such tools are not suitable for use in workshops during the concept phase.

ECML is a modeling language for use in the concept phase. ECML is used by a German automotive company. ECML models consist of simple language constructs and are easy to create with Microsoft Visio (cf. Section 5.4.2). ECML models can be mapped to SysML (cf. Section 5.4.2). SysML Internal Block Diagrams (IBDs) are suitable for modeling items in SysML (cf. Section 2.2.5.3).

The tables in Section 5.5.4 were created in the course of an investigation of numerous product descriptions of safety-relevant components and their relationships. I use these tables in my work to create items with realistic components and component relationships.

In my industrial project with the car manufacturer, ECML was used in conjunction with Microsoft Visio in the concept phase. For the detailed system design, SysML was used in conjunction with Cameo Systems Modeler. Since the transformation of models between two different languages and tools is time-consuming and error-prone, a prototype had to be developed that solves this problem. The development of the prototype was led by me (cf. Section 5.4.5). The following tool is used to document requirements:

- SysML requirements in table form (cf. Section 2.2.5.6)

Based on the SysML IBD, which is generated from the ECML model, the requirements from Phase 1 are refined. This reveals ambiguities in the modeled IBD and improves the understanding of the IBD for people who were not involved in the modeling.

Result of Phase 4

The result of Phase 4 is the item definition [WP-09-01]. The item definition represents a section of the system architecture. The architecture contains only those components and component relationships that are necessary to realize the use cases from Phase 1. For better comprehensibility for external parties, the item definition is specified by requirements.

5.9.5 Phase 5: Security analysis at system level

[WP-15-05] Attack paths

According to ISO/SAE 21434, the identified threat scenarios (from Phase 3) must be expanded into attack paths. The set of attack paths is [WP-15-05]. An attack path consists of individual attack steps. An example of an attack path would be the steps necessary to trick a vehicle's traffic sign recognition system.

Individual attack steps can occur in several attack paths. Furthermore, attack paths can also be part of larger attack paths. Creating copies of attack steps and attack paths in different contexts has the potential for error, as changes must always be made in all copies. This unnecessarily makes conducting an impact analysis difficult.

In this approach, I proceed as follows: I model attack steps as separate and reusable elements in order to facilitate reuse. I combine multiple attack steps into attack paths. I use the following tools to model attack paths:

- Fault Trees (cf. Section 2.3.2).
- SysML requirement diagrams (cf. Section 2.2.5.5)

Since attack paths can overlap in single attack steps, I use attack trees to obtain a comprehensible representation. Attack trees can be extended to fault trees by the additional use of logic gates. Using logic gates (e.g., AND-gate, OR-gate), the cause-effect relationship between attack steps of different levels can be modeled. A Transfer-gate allows for establishing a connection to a subtree, enabling the reuse of already modeled subtrees.

In my approach, I use SysML requirement diagrams to model fault trees. This means that I do not need to develop a separate modeling tool. Furthermore, using the same modeling language (and the same MBSE tool) ensures traceability of already modeled elements from earlier phases.

[WP-15-06] Attack feasibility ratings

According to ISO/SAE 21434, attack paths have to be assessed with regard to their feasibility [WP-15-06]. I use the following tools to assess attack feasibility:

- Attack potential-based approach (cf. Section 2.3.3)

- SysML requirement diagrams in table form (cf. Section 2.2.5.6)

The standard proposes the attack potential-based approach as the first choice for assessing attack feasibility. In this approach, an Attack Feasibility Level is determined from factors such as Time Required, Expertise, Knowledge of the Item or Component, Time Window, and Equipment using a mapping table. For example, suppose the traffic sign recognition of a vehicle can be tricked by displaying a traffic sign on a digital signage system (DSS). The DSS could be manipulated by a cloud attack or by physical access. With the help of the attack potential-based approach, it is possible to determine which attack is easier to carry out and therefore more dangerous.

In my work, I have realized this approach as follows: I take advantage of the fact that the set of attack paths has been represented in the form of Fault Trees using SysML Requirement diagrams. This allows the attack paths to be represented in the form of SysML Requirements in table form. Since attack steps, logic gates, and transfer gates are represented as requirements, they can be listed in the requirements table. At the same time, by modeling the attack paths as Fault Trees, these requirements are automatically arranged in the Requirements Table in the form of a tree structure. I take advantage of the fact that requirements can be extended by attributes. I add the factors of the attack potential-based approach to the requirements and the possibility to define an Attack Feasibility Level in the form of attributes.

If an attack step is part of several attack paths or Fault Trees, the assessment is automatically taken over for the other attack paths or Fault Trees.

[WP-15-07] Risk values

According to ISO/SAE 21434, the risk value for each threat scenario [WP-15-07] is determined from the impact of the associated damage scenarios and the feasibility of the associated attack paths. In general, the following applies: Risk = Impact x Feasibility.

[WP-15-08] Risk treatment decisions

For each threat scenario, at least one risk treatment option [WP-15-08] has to be selected, taking into account the determined risk values: (1) Avoidance of the risk by removing the source. (2) Risk reduction by using a countermeasure. (3) Sharing the risk, e.g., by transferring the risk to an insurance company. (4) Reasonable retention of the risk.

[WP-09-03/04] Cybersecurity goals/claims

Depending on the risk analysis, cybersecurity goals [WP-09-03] and cybersecurity claims [WP-09-04] must be defined in accordance with ISO/SAE 21434.

A cybersecurity claim provides a justification for keeping or sharing a risk. These claims

are retained during the further development process for monitoring purposes. Also, further insights from later development phases may lead to a change in the decision basis.

A cybersecurity goal is a concept-level cybersecurity requirement associated with one or more threat scenarios. An example of a cybersecurity goal is ensuring the integrity of a vehicle's traffic sign recognition system.

If the risk treatment decision for a threat scenario involves risk reduction or risk avoidance, one or more cybersecurity goals must be defined.

[WP-09-05] Verification report for cybersecurity goals

I use the following tools for risk assessment, deriving risk treatment options, describing cybersecurity goals/claims, and documenting the verification report for cybersecurity goals:

- Cybersecurity Assurance Level (CAL) (cf. Section 2.3.4)
- SysML Requirement diagrams in table form (cf. Section 2.2.5.6)

To classify different levels of security in the automotive sector, the concept of Cybersecurity Assurance Levels (CALs) is introduced in ISO/SAE 21434. But its usage is optional. A CAL determines with which level of rigor security activities have to be conducted. CALs can be applied to the entire product life cycle and supply chain. I use CALs because they facilitate appropriate communication about the rigor of security measures along the product life cycle.

For this, certain work products have to be checked with regard to certain verification criteria. For my approach, this means the following: (1) The results from the impact analysis and the feasibility analysis (Phases 2,3,5) have to be checked against the item definition (Phases 1,4) for correctness and completeness. (2) The results from the impact analysis and the feasibility analysis have to be checked against the risk treatment decisions (Phase 5) with regard to completeness, correctness, and consistency. (3) The cybersecurity goals and cybersecurity claims (Phase 5) have to be checked with regard to completeness, correctness, and consistency in relation to the risk treatment decisions. (4) The cybersecurity goals and cybersecurity claims have to be checked for consistency in relation to the item definition.

To do this, I use a digital checklist. For this, I use a SysML Requirement table. This references all work products of the concept phase and assigns the verification criteria to be considered to these work products as attributes.

The digital checklist can serve as a starting point for an auditor. The auditor can (randomly) check whether the referenced work products meet the verification criteria. With the help of the end-to-end linking of the work products and their partial results, design decisions, and the results of the risk analysis can be traced through all work products from the concept phase.

[WP-09-06] Cybersecurity concept

According to ISO/SAE 21434, cybersecurity requirements [WP-09-06] for the item and its operational environment have to be described for the defined cybersecurity goals. The standard requires the description of cybersecurity controls that serve to fulfill the cybersecurity goals. A cybersecurity control is a security measure that helps to prevent cyber attacks or minimize the risk of an active attack. An example of a cybersecurity control is sensor fusion. In the context of traffic sign recognition, sensor fusion combines data from different sensors, such as a camera sensor and radar sensors, in order to enable a more robust and accurate recognition of traffic signs.

The following tools are used for the cybersecurity concept:

- UN R155 mitigations (cf. Section 2.1.1)
- Initial Security Design Pattern Catalogue for the Concept Phase (cf. Section 5.7)
- Approach to applying Security Design Patterns in the concept phase (cf. Section 5.6)
- Approach to deriving requirements from models (cf. Section 5.8)
- SysML Requirement diagrams in table form (cf. Section 2.2.5.6)

UN R155 lists 24 cybersecurity controls that have to be considered in the context of the approval of a vehicle manufacturer's CSMS[3]. The cybersecurity concept is formed from the cybersecurity requirements of the item and its operational environment with associated information about cybersecurity controls.

In the approach, I use a SysML Requirements Table with specific attributes to describe the cybersecurity requirements: (1) Based on the cybersecurity goals, I identify the relevant cybersecurity controls of UN R155. This is done by referencing a requirements table that contains all cybersecurity controls of UN R155. (2) Since the cybersecurity controls of UN R155 only contain a brief description, I also use security design pattern catalogues, which contain detailed information on the design of the cybersecurity controls. (3) After analyzing the security design patterns, I identify the relevant vehicle components in which the cybersecurity control measures have to be implemented. In Section 5.6 I describe how such Security Design Patterns can be realized. (4) Based on this, I derive the textual description of the cybersecurity requirements. I proceed in the same way as in Section 5.8. I use the elements identified to that point to formulate the requirements. Here, it is the identified components that are to receive a countermeasure.

[3] The UN R155 refers to these cybersecurity controls as mitigations. Since the mitigations also include preventive security measures, e.g. the use of access control techniques, I use the term cybersecurity controls here

[WP-09-07] Verification report of cybersecurity concept

According to ISO/SAE 21434, a verification report for the cybersecurity concept [WP-09-07] has to be created. For this purpose, the cybersecurity requirements have to be checked against the cybersecurity goals with regard to the verification criteria completeness, correctness, and consistency. In addition, the consistency of the cybersecurity requirements has to be checked against the cybersecurity claims. Similarly to the verification report for the cybersecurity goals, a digital checklist is used for this purpose.

Result of Phase 5

The main result of Phase 5 is the cybersecurity concept [WP-09-06]. For this purpose, attack paths are modeled in the form of fault trees [WP-15-05] and evaluated with the help of the attack potential-based approach [WP-15-06]. The attack paths are derived from the threat scenarios of Phase 3. The risk values [WP-15-07] are determined based on the impact ratings [WP-15-04] from Phase 2 and the evaluation according to the attack potential-based approach. Depending on the risk values, the risk treatment decision [WP-15-08] is derived. Cybersecurity goals [WP-09-03] and cybersecurity claims [WP-09-04] are determined for high and low risks. A verification report is prepared for the cybersecurity goals [WP-09-05]. For this purpose, several work products are examined with regard to various verification criteria and compared with the defined cybersecurity goals. For risk mitigation, cybersecurity controls are selected and assigned to components of the item defined in Phase 1 and Phase 4. This is described in the form of cybersecurity requirements. Finally, a verification report for the cybersecurity concept is created [WP-09-07].

6 Evaluation

As presented in the section on the research method (cf. Section 3), I have based my work on four iterations. In this Section I present the evaluation of my work for these iterations. In each evaluation iteration, I describe my experiences, findings and conclusions that I used to improve my work for the next iteration. In Section 6.3 I present the implementation of my final approach. The result of my work is a procedure for the development of a cybersecurity concept according to ISO/SAE 21434.

In the Section 6.1 I present the first evaluation iteration. The first iteration served as a general introduction to the field of Model-Based Systems Engineering. Over a period of about 1.5 years, I was allowed to conduct 11 MBSE workshops, mostly with participants from industry. Each workshop took place during the concept phase. I used the workshops to check to what extent the security aspect can be integrated into such workshops.

In the Sections 6.2 and 6.3 I present the second evaluation iteration. I tested the partial solutions I developed at that time with 30 and 67 Master's students, respectively, in a workshop as part of a summer school and in an 8-week project as part of teaching. I tested the following approaches: The approach from Section 5.2, to identify and model damage scenarios using a 3D environment. The approach from Section 5.5, to identify and model threat scenarios using SysML models. The approach from Section 5.6, to resolve threats in SysML models. The approach from Section 5.8, to derive security requirements from SysML models.

In Section 6.4 I present the third evaluation iteration. In the context of an 11-week project in the context of teaching, I tested the partial solutions I had developed up to that point with 140 Master's students. The approaches from the Sections 5.5, 5.6 and 5.8 were reused. The focus of the evaluation was on the use of an initial security design pattern catalogue (cf. Section 5.7).

In Sections 6.2 and 6.3 I present the final evaluation. This iteration focuses on the evaluation of my work with automotive experts from industry and research. Section 6.2 deals with the following: The final evaluation of the approach from Section 5.2, to identify and model damage scenarios using a 3D environment. The evaluation of the approach from Section 5.3, for the use of statistical data for the assessment of damage scenarios. The evaluation of the approach from Section 5.4, for model transformation of ECML models into SysML models. In Section 6.3, I present the implementation of my final approach in a professional MBSE tool.

In the course of three workshops held over the course of a year, I had the opportunity to present my work to experts from the field of automotive security engineering. The work was illustrated and discussed using a continuous application example. The credibility of the work by the subject matter experts was strengthened by a real-life test with a test vehicle.

In Section 6.7 an evaluation of my work takes place on the basis of the requirements from Section 2.5. In Section 7 I address the limitations of my work and derive the need for further research.

6.1 Evaluation 1: Conducting initial workshops

In this section, I report on MBSE workshops that I conducted in the early stages of my dissertation as part of my teaching and industrial activities. The main purpose of the workshops was to teach and apply MBSE content in the concept phase. Usually the work in the concept phase is done by experts from different disciplines. Basically, there were no participants in any of the workshops who had significant security knowledge. I used the workshops to see how security could be integrated into workshops with non-security participants. In contrast to the short description of the workshops in a scientific paper I wrote [**Jap20**], I will report in more detail on my experiences and findings from the workshops in the following. These workshops were the starting point for the work and results of the second evaluation iteration (cf. Sections 6.2 and 6.3).

6.1.1 Workshops characterization

A1	Application scenarios
A2	Visualization
A3	Environment model
A4	Function model
A5	Architecture model
A6	System behavior model
A7	System requirements

Which product aspects were considered and to what degree?

Considered | Partially c. | Not considered

Participant area	Number of workshops	Ø-Participants	Product aspects A1	A2	A3	A4	A5	A6	A7
University	1	8	■	■	■	■	■		■
Management consultancy	2	12	■	■	■	■	■		■
Mechanical and plant engineering	1	25	■	■	■	■	■		■
Radar technology	2	11	■	■	■	■	■	■	■
Farming	5	7	■	■	■	■	■		■

Figure 6-1: Conducted MBSE workshops

Over a period of 1.5 years I conducted 11 workshops on MBSE (cf. Figure 6-1). These workshops were at the level of the conceptual phase. The workshops took place in presence. My role was to prepare and present the content and to moderate. For modelling, several boards with brown paper or several whiteboards were used. Additionally, cards representing model elements of the modelling languages CONSENS or SysML were used. In general, workshop participants were divided into several teams. The results were then presented to the other teams. This enabled ambiguities to be identified and suggestions for improvement, e.g. in the modelling, to be incorporated.

The first workshop was a workshop with master students. This was followed by 10 work-

shops with customers from industry. In all workshops, participants came from different disciplines. However, most of them were from industrial and mechanical engineering and some from computer science and business informatics.

In general, the CONSENS method (cf. Section 2.2.3) was used in all workshops. Figure 6-1 shows which product aspects of CONSENS were considered and to what extent. The CONSENS modelling language was used in almost all workshops. In the workshops with participants from the radar technology area, the modelling language SysML (cf. Section 2.2.5) was used.

The CONSENS method does not consider the security aspect. As a possible extension of the CONSENS method, I have investigated to what extent CONSENS can be extended by the security aspect.

6.1.2 Details about the workshops

In the following I will describe the workshops in more detail. I will also report on the experiences I was able to gather regarding the consideration of security in the workshops.

Through discussions with industry participants, I learned that they relied on the following: consideration of security in the concept phase was not considered relevant. Participants relied on security experts to make everything secure in later phases of development. This approach misses the potential to identify and fix identifiable security vulnerabilities in the system design at an early stage. This would result in an enormously high adaptation effort for errors identified at a later stage.

The following product aspects were covered in the workshops with participants from the university, management consultancy and mechanical and plant engineering: The main focus of the training was MBSE. The training contents were presented and elaborated on the basis of an application example. The application example was the programmable toy robot COZMO [**DDL22-ol**]. The participants had to create a concept for an extension of COZMO. The goal of the extension was to enable several COZMO robots to play football together. Use cases were identified (cf. Figure 6-1, Aspect A_1). A COZMO robot was used to visualise and demonstrate existing functions (A_2). LEGO bricks were used to visualise and discuss possible extensions using LEGO Serious Play (LSP) [**LSP22-ol**]. LSP is a method to solve complex problems and generate innovative ideas with the help of LEGO bricks. With LSP, complex topics can be processed and communicated visually and haptically. The participants had the task to extend an existing functional model with possible new functions (A_3). The participants found that it was unclear what the operational environment had to look like (A_4). For example, it needed to be clarified what should be used as a football and what as a goal and how many COZMOs should play in a team. In order to realise the football game, the participants found that the gripping mechanism had to be adapted (A_5). In addition, the participants found that the visual detection of other COZMO robots was not sufficient due to the low camera angle. The majority

of participants supported the use of Bluetooth as an additional communication channel between the COZMO robots. A system behaviour model was not created (A_6). At the end of the workshops, requirements (A_7) were derived based on the created product aspects. COZMO can be controlled by a smartphone. The video signal can be displayed on the smartphone. Displaying the video signal on a non-authorised smartphone was identified as a critical privacy issue. Participants identified the remote control of the COZMO as security relevant if unauthorised smartphones could perform the control. The security threats were noted in the form of threat cases (A_1). In addition, the assets were marked in the system architecture (A_5). These were either components or component relationships. Although the participants were able to identify the critical elements, they were not able to resolve the security vulnerabilities at the concept phase level.

In the workshops with participants from the field of radar technology, the goal was to create a concept for an autonomous vehicle capable of driving in a platoon. In contrast to the other workshops, the modelling language SysML was used instead of CONSENS. For A_2 no visualisation was created by the participants. Regarding A_6, the use cases of the autonomous vehicle were modelled using several behaviour diagrams. In general, the workshop participants were able to identify security critical elements for all product aspects except A_2. In these workshops, the participants were not able to resolve the security vulnerabilities at the concept level.

The workshops with participants from the agricultural sector involved the design of a real sensor system for a corn harvester. Use cases for the sensor system to be developed were identified (A_1). A physical model of a corn harvester on a scale of 1:32 was used for visualisation (A_2). In particular, the attachment of the sensor system to the corn harvester was discussed. Due to the expertise of the participants, no further visualisation was used. At the level of the environment model, elements interacting with the sensor system were identified (A_3). Two application environments for the sensor system were identified. One was the use of the sensor system in the corn harvester and the other was the use of the sensor system in the laboratory. This resulted in all product aspects being created separately for the two application environments due to the high complexity of each application environment. As a result of this process, it was identified that an additional power supply was required for the laboratory environment. Functions were identified and grouped for both application environments (A_4). In contrast to the CONSENS functional model, no functional structure was created. To realise the sensor system, the necessary components and component relationships were identified and aggregated in an architecture model (A_5). Based on the created product aspects, the requirements for the realisation of the sensor system were derived from the participants (A_6). For each product aspect I asked the participants which elements could be security critical. At the level of the system architecture model (A_5), one component was identified as an asset containing licensing information. This component determined the criteria by which the data collected by the sensor system would be analysed and evaluated. By modelling different application environments (corn harvester and laboratory) it was identified that the use of the sensor

system can be licensed depending on the application environment. This again identified the licensing component as an asset. In these workshops, the participants were not able to resolve the security vulnerabilities at the concept level.

6.1.3 Lessons learned

The workshops taught me how to apply MBSE in the concept phase. It also gave me a sense of what topics can be worked on and how much time is available. Basically, even simple tasks take a lot of time to work on because the workshop participants have to agree with each other. Participants have different views and experiences on the same aspect. The discussions in the workshops improve the common understanding. Ambiguities can be identified and resolved in the workshop. This reduces the redesign effort in later development steps. The use of models in such workshops is limited. This is because the participants come from different disciplines. For example, a mechanical engineer does not usually understand detailed SysML diagrams and a computer scientist does not usually understand how to read a technical drawing and what aspects need to be considered in it. The only suitable means of communication in such workshops are simple model constructs.

The problem with discussions is that the general level of granularity of the concept phase is not maintained. In this case, discipline-specific details slow down the process of building overall understanding. Maintaining the level of granularity is, on the one hand, the task of the moderator. On the other hand, a clear task definition with appropriate model constructs also supports maintaining the level of granularity.

In general, the identification of vulnerable components and threat cases was done in all workshops with the non-security participants. From a methodical point of view, it was only necessary to ask the participants to identify vulnerable or security-critical elements in each product aspect. Incorporating the security aspect was mainly a matter of moderation and questioning on the part of the moderator. The workshop participants have the domain knowledge. The combination of the moderator's security-related questions and the workshop participants' expertise enables the identification of vulnerable or security-critical elements.

Color coding or the use of stereotypes could be used to extend the modelling language for marking security-critical elements.

Although the participants discussed possible countermeasures, none of the workshops succeeded in eliminating the vulnerabilities due to a lack of knowledge. In addition, I learnt that changing the original product aspects in order to apply a countermeasure is an enormous effort for the workshop participants.

Based on my experiences from the 11 workshops, I have developed the following approaches: In Section 5.5 I present an approach for identifying security threats. This approach uses only simple model constructs, so no in-depth knowledge of MBSE is required to understand and apply these constructs. In Section 5.6 I present an approach that

supports the resolution of security threats using security design patterns in early system design. Only simple model constructs have been used for this purpose. In order to maintain the level of granularity, to ensure easier applicability in the concept phase, and for the general understanding of the participants, discipline-specific details have been largely omitted.

6.2 Evaluation 2: A - Dortmund International Summer School

In this section I present the evaluation of the approach from Section 5.5. The topic of the paper is the identification of threats in workshops using models. The approach and the evaluation were presented in [**JAD21**].

6.2.1 Project characterization

In the context of the Dortmund International Summer School [**Dor20-ol**] in the track Automotive (Systems) Software Engineering I presented the approach in a one day workshop with 30 international master students from the fields of computer science and computer engineering. The approach was applied from a systems engineering perspective by 6 different teams using platooning as an application example. The results of the workshop were evaluated by the track leader in three further workshop days and reused to derive testable software requirements using the SCIL approach [**WJK+20**]. I had previously tested the approach in two internal workshops with three participants each.

6.2.2 Evaluation goal

The aim of the evaluation was to check the applicability of the approach in workshops and to find potential for improvement for future work. In particular, I wanted to find out which activities/tools/constructs needed to be adapted in order to increase the effort/benefit ratio of using the approach.

6.2.3 Evaluation results

Using Figures 6-2 to 6-4 I will explain the results. In Activity 1 (cf. Figure 6-2), each team created an environment model in the form of a SysML IBD and, on median, 2 threat scenarios in the form of SysML sequence diagrams within 2 hours. The 3D environment presented in Section 5.2 was used to visualise the damage scenarios caused by the threat scenarios. Stereotypes were used to highlight identified security-relevant functions and provided a basis for joint discussion within the teams. The functions formally corresponded to messages in a SysML sequence diagram. On median, the teams identified 11 (78 %) security-relevant and 4 (22 %) potentially security-relevant functions, distributed across all threat scenarios.

In Activity 2 (cf. Figure 6-3), which lasted 6 hours, each team created more detailed

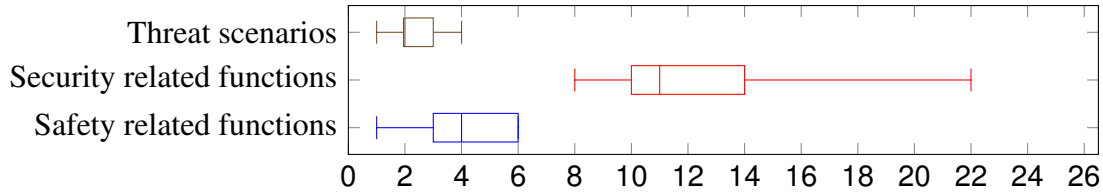


Figure 6-2: Activity 1 - Identified security and safety relevant functions at the vehicle environment level.

threat scenarios at the system architecture level. In this context, the component table (cf. Figures 5-16 - 5-18) allowed each participant to fill in missing knowledge about functions in unknown vehicle components and their relationships to each other. On median, 19 (86 %) security-relevant and 3 (14 %) potentially safety-relevant functions were identified across all detailed threat scenarios at the system architecture level. A median of 4 (17 %) security critical and 2 (6 %) safety critical functions were identified across all detailed threat scenarios at the system architecture level.

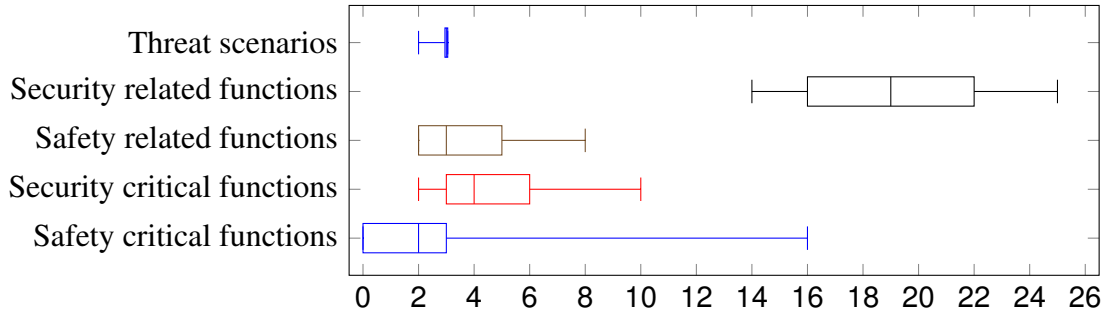


Figure 6-3: Activity 2 - Security and safety relevant functions identified at vehicle architecture level.

In Activity 3 (cf. Figure 6-4), which lasted 1 hour, each team derived a system architecture model in the form of a SysML IBD. This was done using the detailed threat scenarios from Activity 2, which were available in the form of SysML sequence diagrams. The objects of the sequence diagrams were mapped to the components of the system architecture model. On median, a system architecture consisted of 12 components. The functions of the sequence diagrams were mapped to component relationships. Identical functions may be contained in more than one sequence diagram. Thus, in the described mapping from a set of sequence diagrams to a system architecture, the number of component relationships was less than or equal to the total number of functions. On median, a system architecture consisted of 13 component relationships. Taking into account the functions from Activity 2 that were rated as critical, the median number of component relationships in the created architecture model was 4 (31 %) security-critical and 1 (6 %) safety-critical.

6.2.4 Summary of results

On median, initially in Activity 1, 78% (or 22%) of the functions were relevant regarding security (or safety). Here the threat scenarios were modelled using SysML sequence

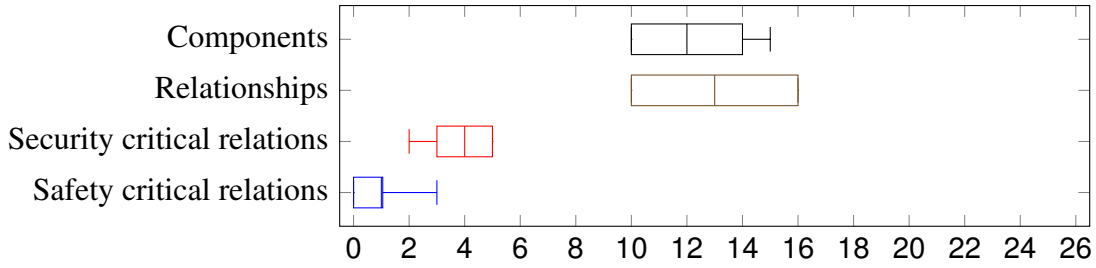


Figure 6-4: Activity 3 - Security and safety related elements in system architecture.

diagrams. In Activity 1, the information is not very detailed and no distinction is made between critical and less critical functions. Activity 2 provided a level of detail for the threat scenarios. On median, after applying Activity 2, 17% (or 6%) of the functions were security (or safety) critical. At this point, the critical functions are spread across multiple threat scenarios. Activity 3 allowed the threat scenarios to be aggregated into an initial system architecture. This made it possible to locate the security or safety critical component relationships. On median, after applying Activity 3, 31% (or 6%) of the component relationships were security (or safety) critical.

6.2.5 Lessons learned

Overall, the approach supported the refinement of threat scenarios and the localisation of critical component relationships in the initial system architecture during the concept phase. During the workshop, all 6 teams were able to complete all activities. Using the models, the team members were able to discuss security and safety issues in the workshop and resolve them by adapting the models. The approach of first creating the threat scenarios and then using them to create an initial system architecture was helpful in reducing the amount of work required of the teams. In general, it was time consuming for the participants to determine the safety ratings using the two approaches. This effort could be reduced by a support tool, e.g. in the form of a formula-based table. A Microsoft PowerPoint template was provided for modelling. On the one hand, several participants could carry out the modelling in parallel, and on the other hand, the training effort was low. On the one hand, an online SysML editor that could be used by several users at the same time would improve modelling. However, this often requires licences and the training effort is higher. Due to the tight timeframe, it was necessary for the teams to work on the threat scenarios in parallel. The threat scenarios were first discussed in general in the individual teams, then detailed and modelled in subteams of 1 or 2 people, and finally discussed and refined together in the individual teams.

6.3 Evaluation 2: B - MBSE 2020

In the context of teaching at the University of Paderborn in 2020, I was allowed to develop and carry out a project accompanying the lecture Model-Based Systems Engineering. [1] The project was based on four of my approaches (cf. Sections 5.2, 5.5, 5.6 and 5.8). In this section I report on the evaluation of this project. These evaluation results are based on two scientific papers [**JA21**; **JAK+21**]. The results of the teams were summarised in slides. A total of 658 slides were quantitatively analysed for this evaluation. As for the qualitative aspects, the statements are based on the weekly videoconferences I held with the students to clarify unclear points.

6.3.1 Project characterization

As part of a project at the University of Paderborn in 2020, I tested the approaches with 67 interdisciplinary master students from the fields of computer science, computer engineering and business informatics. The project was carried out using platooning as an example and lasted 8 weeks with 80 hours per student (total workload 5360h). The project was conducted virtually due to the COVID-19 pandemic.

The project consisted of the following activities (Activity 1) Identification and visualisation of application scenarios and security-related damage scenarios (threats). (Activity 2) Model derivation and risk analysis. (Activity 3) Selection and application of countermeasures. (Activity 4) Derivation of security requirements.

I developed the individual activities in advance and tested them with a team of three to identify and eliminate initial ambiguities. In order to achieve comparable results between the teams, a minimum number of models and requirements per team had to be created. In order to determine an appropriate minimum number and to find initial ambiguities in the application of the approaches, the test team applied each of the four activities in advance.

Each week I held a one-hour video conference to check on the status of the project and discuss ambiguities in the application of the approaches. In total, 21 teams of about three people each used the approaches to identify threats in early system design, select and apply countermeasures, and derive security requirements. On average, each team included students from different courses, so most teams were interdisciplinary.

Each team had to choose an application area in the automotive context. The choices were autonomous parking, autonomous city driving and platooning. Most teams chose autonomous parking and autonomous city driving. I created an evaluation sheet with different criteria that was used to assign roles within each team. To address the students' lack

[1] Consideration of the data protection ethics of the University of Paderborn: The results of the students were evaluated and anonymised. There were no objections to the use of the results. In particular, the students were informed that an objection to the sharing of the results would have no negative consequences for these students.

of knowledge about functions, components and component relationships in autonomous driving, the students were asked to use the component cheat sheet from Section 5.5.

The individual team workshops were conducted virtually in Microsoft Teams using Microsoft PowerPoint as the modelling tool. I provided SysML templates for PowerPoint. For the requirements definition, I provided spreadsheet templates for Microsoft Excel.

6.3.2 Evaluation goal

The goal of the project was to apply the approaches described in Sections 5.2, 5.5, 5.6 and 5.8 in a consecutive project.

The project was meant to test the applicability of the approaches in workshops, and to identify possible improvements for future work. In particular, through the project I wanted to find out which activities/tools/constructs need to be adapted to increase the effort/benefit ratio of applying the approaches. In addition, I wanted to find out what impact the application of the approaches had on the artefacts created.

6.3.3 Evaluation of the results from Activity 1

In Activity 1 of the project which lasted about 20 hours, the 3D environment 3DE presented in Section 5.2 was used to identify and model application scenarios and security relevant damage scenarios in the context of platooning. A comprehensive evaluation of the approach from Section 5.2 is presented in Section 6.5.1.

At this point, version 1.0 of 3DE was available and was made available to the teams. Based on previous tests of 3DE, I was able to determine that a fixed library of 3D objects was not sufficient to model as many different scenarios as possible. Also, the use of abstract and nameable dummy objects (e.g. a cube with a label called “router”) was not helpful for visualisation. 3DE has been enhanced for version 1.0 as follows: (1) If a 3D object for visualisation is missing when modelling an application and a damage scenario, it is possible to import 3D models that are (freely) available on the Internet. This work was carried out by student assistants under my supervision. (2) If no suitable 3D object can be found, it should be possible to create a 3D object in a short time. For this purpose the tool Cubes was realised. Cubes was developed as part of a student project [**Sch20**] supervised by me. Cubes allows the creation of 3D objects based on voxels. Version 1.0 of 3DE was used as the basis for a scientific paper I wrote on the use of a 3D environment for domain knowledge elicitation as part of the concept phase [**JKK20**].

Using 3DE, the teams were able to identify on median 5 application scenarios and 5 security-related damage scenarios (cf. Figure 6-5).

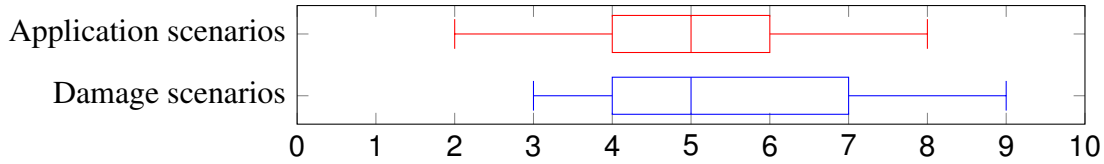


Figure 6-5: Identified application and damage scenarios from Activity 1.

6.3.4 Evaluation of the results from Activity 2

In Activity 2 of the project, which lasted about 20 hours, models were derived based on the identified use case scenarios and security-relevant damage scenarios. A risk analysis was also carried out. For this purpose, the approach described in Section 5.5 was used.

The evaluation of the approach has already been presented in Section 5.5 and in the scientific paper [**JAD21**].

In Activity 2, models are created at the level of the vehicle environment (black box model) and at the level of the vehicle architecture (white box model). These models are not complete, but represent only a subset of the model elements necessary to implement the application and damage scenarios identified in Activity 1. The vehicle environment model is represented by a SysML IBD (B-IBD) and the behaviour at this level is represented by SysML sequence diagrams (B-SD). The vehicle architecture level models use a SysML IBD (W-IBD) extended by components and component relationships. System behaviour at this level uses SysML sequence diagrams (W-SD), which use the components and component relationships of the extended SysML IBD (W-IBD).

On median, a system architecture consisted of 17 components and 21 component relationships (cf. Figure 6-6).

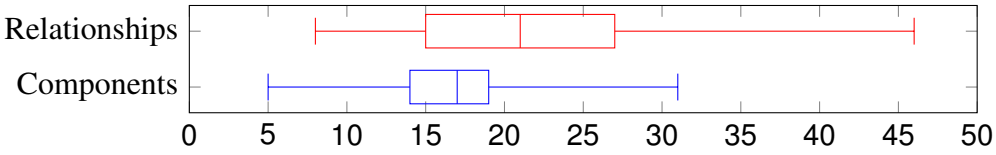


Figure 6-6: From Activity 2, relevant components and relationships.

6.3.5 Evaluation of the results from Activity 3

Activity 3 of the project lasted about 20 hours and was carried out together with Activity 2.

The B-IBDs created in Activity 2 and the B-SDs created were used to derive requirements at the vehicle environment level (B-REQs). The W-IBDs created in Activity 2 and the W-SDs created were used to derive requirements at the vehicle architecture level (W-REQs). As models were created at the vehicle environment level (B-IBDs, B-SDs) and initial requirements were derived from them (B-REQs), the additional creation of models at the system architecture level (W-IBDs, W-SDs) caused extensive changes to the requirements at this level (W-REQs).

B-REQs referring to the B-IBD were adapted to (K1) 55% (median). B-REQs referring to the B-SDs were adapted to (K2) 60.5% (median). The creation of the W-IBDs and W-SDs resulted in additional new requirements. (K3) 15% (median) of the W-REQs related to the W-IBD were new. (K4) 9% (median) of the W-REQs related to the W-SD were new.

I interpret the key findings as follows The percentages in K1 and K2 are so high because the creation of the white box model (W-IBDs, W-SDs) helped to create knowledge about components, component relationships and component functions, so that white box requirements (W-REQs) could be formulated more precisely. By creating the white box model, in addition to knowledge about the components, a better overall understanding of the system under development was gained, so that new requirements could emerge K3 & K4.

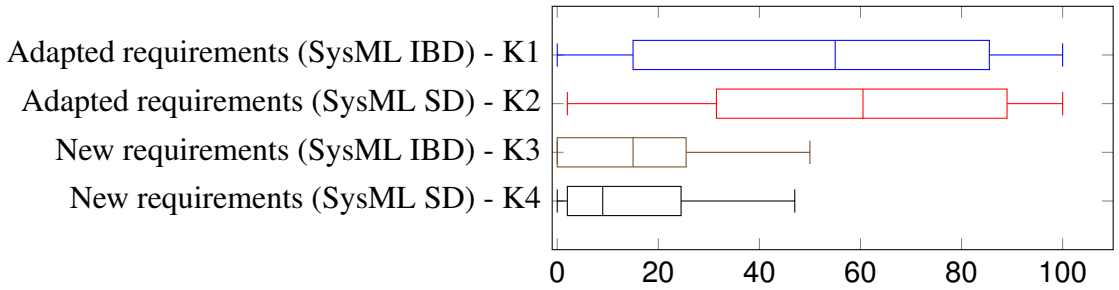


Figure 6-7: Impact of created white box models on initially created requirements.

6.3.6 Evaluation of the results from Activity 4

In Activity 4 of the project, which took about 20 hours, countermeasures were selected and applied based on the risk analysis from Activity 2. This was done using the approach described in Section 5.6. The security design patterns from Section 5.7 were provided for this purpose. At that time, few of the security design patterns mentioned in Section 5.7 existed. In Section 6.4, I report on the use of all 10 security design patterns in a later project.

Figure 6-8 quantitatively describes the impact of applying the countermeasure on the system architecture. On median, 2 new components and 4 new component relationships were added by the countermeasure. At the same time, on median, 0 components and 1 component relationship were changed. Compared to Figure 6-6, the application of a countermeasure on the median caused the following (1) 19 % of the component relationships and (2) 5.9 % of the components were new. (3) 9.5 % of the component relations and (4) 0 % of the components were changed.

6.3.7 Evaluation of the overall project

The teams had the opportunity to improve their final MBSE exam grade by performing well or very well in the accompanying project.

During the project, the teams had to submit several deliverables that were evaluated. To

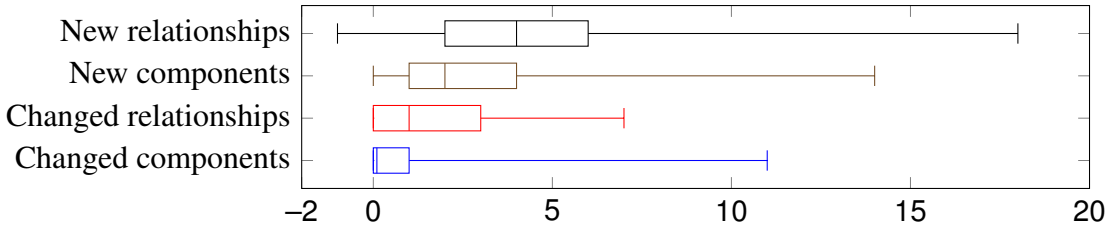


Figure 6-8: From Activity 4, the impact of applying the countermeasure on components and relationships.

take part in the exam, each team had to achieve 50% of the total points. If 70% was achieved, the teams received one grade level (0.33). If 90% were achieved, they received two grade levels (0.66).

On the basis of the marks achieved, I assessed the overall success of the project. Figure 6-9 summarises the grade steps achieved by all teams. 7 teams achieved one level and 1 team achieved two levels out of a total of 21 teams. In total, 40% of the teams achieved at least 70% of the total points. 60% of the teams have only passed the project without having received any grade steps. I rate the overall success of the project as satisfactory, but with room for improvement. The project was repeated the following year with revised content. The students achieved better results (cf. Section 6.4).

	Very good results: at least 90% of the points		Good results: at least 70% of the points		Else: at least 50% of the points	
	Teams	%	Teams	%	Teams	%
MBSE Project 2020	1/21	7	7/21	33	13/21	60

Figure 6-9: Summary of the evaluation of the results of all teams.

6.3.8 Lessons learned

Overall, all teams were able to identify and model use cases and security-related damage scenarios in joint workshops during the project. The teams were able to derive requirements from the models and select and apply countermeasures using the security design patterns provided.

In order to facilitate the work in the team, the team members should each take on one of the roles of security expert, safety expert and modelling expert. During the course of the project, team members should primarily act from the perspective of these roles. The roles were defined after completing a strengths and weaknesses questionnaire. According to the teams, the role concept had no impact on the project work. This was a missed opportunity to carry out an analysis of the students' strengths and weaknesses before the teams were formed and to propose a team composition based on this. I took this into account later in my work (cf. 3rd evaluation iteration in Section 6.4).

I had not explained exactly how to work together in the project, so some teams could not always work effectively. There was an understanding that all participants had to work on one task at a time. This meant that some teams missed the opportunity to work on subtasks in parallel. For example, modelling damage scenarios using SysML sequence diagrams. This led to more precise descriptions of the approaches later in my work.

The given time frame of 80h (approx. 10 working days) for several workshop dates and several participants would be unsuitable in an industrial environment. In my experience, such workshops tend to invite leading and experienced experts who often have many appointments and little time. A time frame of 5 working days would be more realistic for carrying out the activities.

Regarding Activity 1: The use of the 3D environment 3DE helped the teams to identify application scenarios and damage scenarios at the level of the vehicle environment. In addition, the visualisation facilitated communication between participants and the presentation of results to other teams. The usability of 3DE was criticised by several teams. This was improved in subsequent versions of the tool. During the project I found that 3DE was not useful for modelling security-related damage scenarios at the vehicle environment level. Even when security-related 3D objects, such as a router, were integrated into the 3D environment. It turned out that 3DE is primarily well suited for safety-related damage scenarios. Based on this experience, 3DE was used in subsequent projects primarily for the identification and visualisation of damage scenarios.

With regard to Activity 2: All teams were able to use the SysML constructs I provided. The SysML models were created using a Microsoft PowerPoint template that I provided. On the one hand PowerPoint was very easy for the students to get started with, but on the other hand it lacked features to support modelling, such as automatic alignment of multiple model elements. On several occasions the teams told me that modelling more complex models in PowerPoint was time consuming. Applying the risk analysis methods was straightforward using the spreadsheets I provided, but manually determining safety levels for the large number of functions based on student reports was time consuming. Based on this experience, I later ported the SysML constructs and risk analysis into a professional MBSE tool for easier use (cf. 4rd evaluation iteration in Section 6.6).

Regarding Activity 3: The derivation of requirements from models basically worked well. The quality of the model mostly determined the quality of the requirements.

Regarding Activity 4: The choice of security design patterns for the participants was very limited, as I could only provide very few design patterns suitable for the workshop at that time. This led to the creation of an initial security design pattern catalogue later in my work (cf. Section 5.7).

6.4 Evaluation 3: MBSE 2021

In the context of teaching at the University of Paderborn in 2021, I was allowed to develop and carry out a project accompanying the lecture Model-Based Systems Engineering. [2]

The project was partly based on the same activities as the 2020 project (cf. Section 6.3). The project results of all teams were available in the form of slides and tables, which were evaluated quantitatively. As for the qualitative aspects, the statements were based on the weekly videoconferences I held with the students to clarify unclear points.

There were two main improvements compared to the previous year's project: At the beginning of the project, a competence test was carried out, on the basis of which the team composition was determined. In addition, the initial Security Design Pattern (SDP) catalogue presented in Section 5.7 was used. The evaluation of the application of the SDPs was presented in a scientific paper [**JFA+23**]. In the following, I will mainly report on the evaluation of the two improvements.

6.4.1 Project characterization

I conducted the project with 140 master's students. The project was carried out using platooning as an example and lasted 11 weeks. The project was conducted virtually due to the COVID-19 pandemic.

The students were from Computer Science, Business Informatics and Computer Engineering. In total there were 28 teams of 5 people. The students spent a total of 8400 hours on the project.

The project consisted of the following activities (Activity 0) The teams were formed for the preparation. First I gave a one hour competence test. In this test I asked about the areas of security, safety, modelling, requirements engineering and project management that are relevant for the concept phase. Based on this, I proposed team constellations in which the required competencies were represented by at least one person in the team. In addition, the participants had the opportunity to form a team themselves. (Activity 1) In this activity use case scenarios and threat scenarios for platooning were identified. (Activity 2) This activity modelled the relevant components and relationships of a system architecture required to realise the scenarios identified in Activity 1. (Activity 3) In this activity a risk analysis was carried out. Based on this, decisions on how to deal with the risks were defined. (Activity 4) If the risks were high, a countermeasure should be chosen to minimise the risk. The teams could choose between two variants. (Variant A) Selection of countermeasures from the initial catalogue of SDPs described in Section 5.7. (Variant

[2] Consideration of the data privacy ethics of the University of Paderborn: The results of the students were evaluated and anonymised. There were no objections to the use of the results. In particular, the students were informed that an objection to the sharing of the results would have no negative consequences for these students.

B) Selection of countermeasures based on publications available on the Internet. (Activity 5) To verify the effectiveness of the countermeasure, a new risk analysis was performed on the changed system components and their component relationships. (Activity 6) Derivation of requirements.

The activities were tested in advance as part of a master's thesis, which I supervised. Among other things, the master's thesis served to create a continuous application example and to identify and eliminate ambiguities in the activity descriptions in advance [**Fah21**].

6.4.2 Evaluation goal

The goal of the project was to apply the approaches described in Sections 5.2, 5.5, 5.6, 5.7 and 5.8 in a consecutive project.

In particular, two improvements to the previous year's project had to be tested: the application of the competency test (Activity 0) and the application of the initial SDP catalogue (Activity 4). In the following I will concentrate on the evaluation of these two improvements.

Regarding the application of the initial SDP catalogue, I had the following evaluation objectives: (Objective 1) Quantitative comparison of two variants. Variant A allowed the use of countermeasures in the form of SDPs. Countermeasures were selected using a selection table. [3] Simplified, this table allowed a step-by-step restriction of SDPs based on the following questions: (Q1) Does the SDP fit the system/component level under consideration? (Q2) What type of protection is required? In Variant B, countermeasures should be derived and applied from scientific publications publicly available on the Internet. (Objective 2) Evaluate the feedback from the teams of both variants.

6.4.3 Evaluation of the competence test

The activities in the design phase require different skills. In general, modelling and requirements engineering skills are required. As the project lasted several weeks, project management experience and the willingness of at least one team member to lead the project was also required. These skills were also required for the security and safety aspects of the concept phase.

To improve the success of the whole MBSE project, I first conducted a competence test. Based on the test results, a team composition was proposed. In addition, there was the option to suggest a team composition. This option was only used in a few cases due to the COVID-19 pandemic at that time. According to discussions with the students, at that time the students worked almost exclusively virtually and from home and therefore had little opportunity to get to know each other.

---

[3] This table and the assignment of SDPs to it were described in a master's thesis [**Fah21**] I supervised.

The competency test took one hour to complete. The test consisted of 44 multiple-choice questions. The questions were selected from a variety of subject-related sources [**Pew17-ol**; **AST18-ol**; **IRE21-ol**; **Car16-ol**]. The questions on project management related to personal project experience and included whether participants would take on the role of project manager in the team. 75 % of the students agreed to take on this role.

Figure 6-10 shows the evaluation of the test results. On median, the students achieved good results in the area of security (80%) and good results in the area of SysML/UML (70%). For the other skills, the median results were in the middle range. Participants were distributed across teams so that each competency was represented by at least one member with a high score in that competency.

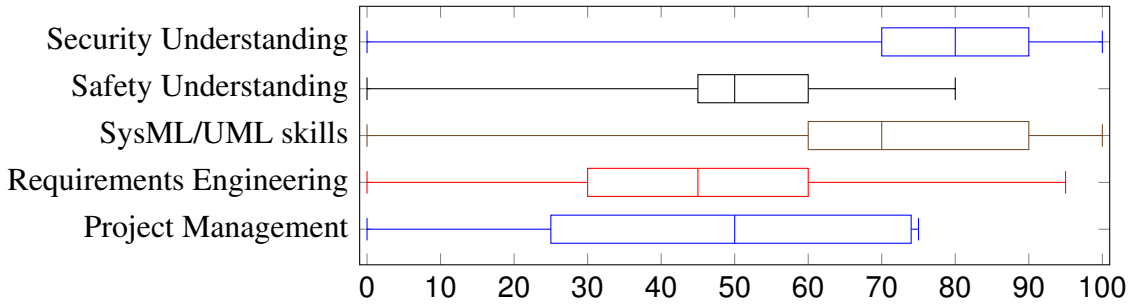


Figure 6-10: Evaluation of the competence test with N = 140 *master students.*

The previous year's MBSE 2020 project (cf. Section 6.3) did not include a skills test. Instead, the teams were randomly assigned. Once the teams had been formed, the participants were asked to complete a strengths and weaknesses profile in relation to various competences and, based on this, to take on a role in the project.

Participants in the MBSE 2020 project reported several times that the random distribution and the role profile did not have a positive impact on the success of the project. In contrast, in the MBSE 2021 project, there was not a single report of poor team composition or lack of skills in the team. I assume that the use of the competency test as a tool for team composition had a positive effect on the overall project results (cf. Section 6.4.6).

6.4.4 Quantitative comparison between two test groups

This section refers to Activity 4 of the MBSE project. Based on a risk analysis, the teams had to select countermeasures to address security-critical vulnerabilities in the architecture. I adopted the procedure for applying countermeasures in the concept phase from [**Jap21**]. This work served as an application example for the teams. The 28 teams were free to choose the variants. Variant A was chosen by $N_A = 18$ teams and Variant B was chosen by $N_B = 10$ teams. To compare the effectiveness of the application of the countermeasures, each team had to record several indicators. In the following, I present the evaluation of the most important indicators regarding the application of SDPs.

Based on the use case scenarios and threat scenarios from Activity 1, models were created in

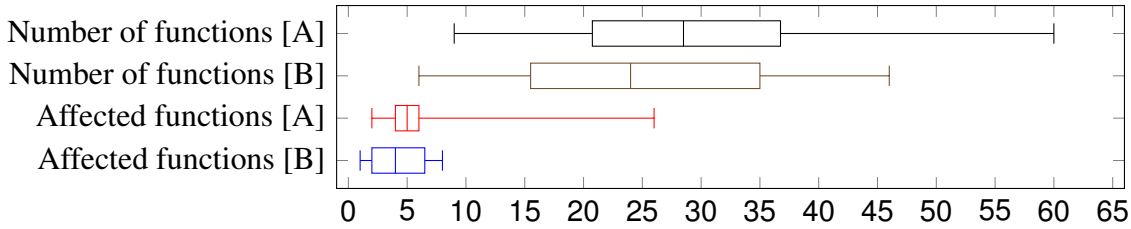


Figure 6-11: Functions affected by the application of the countermeasure.

Activity 2 in the form of an IBD (system architecture) and several SDs (system behaviour). Functions were part of the SDs and represented the relationships between individual components in the IBD. Figure 6-11 compares the number of functions created by the teams in the two variants. A function is triggered in a component/system and has an effect in the same component/system or in another component/system. The application of countermeasures affected a subset of the functions created. At the median, Variant A teams created 28 functions, of which 5 were affected by the median countermeasure. For Variant B, there were 24 functions, of which 4 were affected by the countermeasure. The median is almost identical for both variants before and after applying the countermeasure.

I think that the choice of variant had no clear influence on the number of functions created or the number of functions affected by the countermeasure.

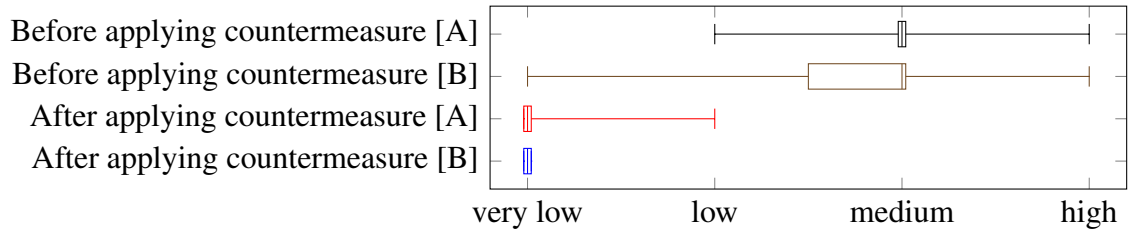


Figure 6-12: Assessing the feasibility of hacking attacks.

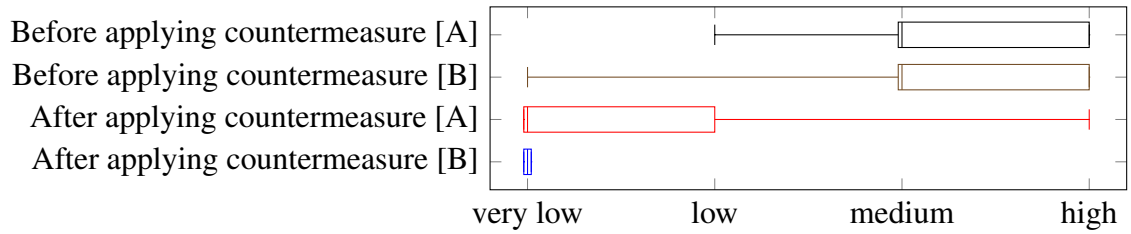


Figure 6-13: Assessing the safety impact of hacking attacks.

Figure 6-12 shows the feasibility rating data for the teams of both variants before and after applying the countermeasures. The safety impact analysis data and calculated risks are shown in Figures 6-13 and 6-14. The median of the feasibility rating, impact analysis and calculated risks were identical for both variants before and after applying the countermeasure.

For the majority of the feasibility rating data, the values were identical before and after the countermeasure was applied. I think that the choice of variant did not have a clear

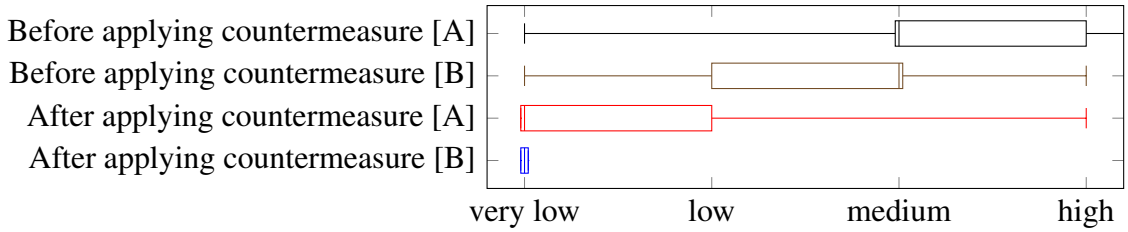


Figure 6-14: Calculated risks.

influence on the feasibility rating.

The safety impact is identical for both variants for 50% of the data. After applying the countermeasure, 50% of the data were in the very low/low range for both variants. The safety impact after application of the countermeasure is very low for Variant B. This could be due to the fact that the number of 10 SDPs for Variant A was too low. In contrast, the number of available publications presenting countermeasures in Variant B is much higher. As a result, more appropriate countermeasures could be selected in Variant B, resulting in a lower security impact.

In order for Variant A to be competitive with Variant B, the initial design pattern catalogue needs to be expanded. The calculated risk differs more for the two variants. Basically, Variant A has a high or medium risk for 50% of the data, which changes to low or very low after the countermeasure is applied. Similarly, for Variant B, the risk shifts from medium or low to very low for 50% of the data.

The risk depends on the feasibility rating and the safety impact. The values of the feasibility rating are almost the same for both variants. The data for Option B are basically lower for the safety impact. As a result, the risk is generally lower for Option B.

6.4.5 Evaluation of feedback

In the following, I summarise the feedback from the 28 teams and derive the next steps (cf. Figure 6-15) [4]. The positive feedback H1-H5 is self-explanatory. Regarding D1, D4, D5, D6: I believe that the reference to detailed information in the form of scientific publications as a supplement to the SDPs will improve this. Regarding D2, D3: The work in the concept phase is characterised by the collaboration of leading experts who often have little time [**Jap21**]. Variant B is not suitable for use in such workshops because the free choice from a very large number of possible countermeasures, described in great detail, requires a lot of time to understand and apply. An expansion of the pattern catalogue might help here.

[4] In total, I received 8 A4 pages of feedback. For high quality positive and negative descriptions the teams could get bonus points for the final exam

	…was difficult	… was helpful
Both	D1: Choice between different possible countermeasures. (3x)	H1: Understanding of own architecture based on Activity 2 to apply countermeasure. (13x)
		H2: Provided application example. (7x)
B: Internet	D2: High effort to understand the sources found and high effort to apply the countermeasure found. (9x)	H3: Abstraction of the descriptions and models of the countermeasures found. (1x)
	D3: High effort to find the appropriate countermeasures. E.g. Initially the descriptions seem to fit. On closer examination, the countermeasures did not fit. (6x)	
A: Pattern	D4: Transition of the design pattern to self-created architecture. (4x)	H4: Description and models of design patterns. E.g. by comparing the SDP descriptions or by checking which elements of the SDP fit to the own architecture. (11x)
	D5: Use of the selection table did not limit countermeasures to exactly one countermeasure. (5x)	
	D6: The description of the design patterns was not sufficient for a deep technical understanding. (1x)	H5: Basically, the selection table was helpful to narrow down the possible countermeasures. (9x)

Figure 6-15: Aggregation of the feedback from the 28 teams.

6.4.6 Evaluation of the whole project

As in the previous year's project, teams had the opportunity to improve their final MBSE exam grade by performing well or very well in the MBSE project.

During the project, the teams had to submit several deliverables that were evaluated. To participate in the exam, each team had to achieve 50% of the total points. If 70% was achieved, the teams received one grade step (0.33). If 90% were achieved, they received two grade steps (0.66).

I evaluate the overall success of the project on the basis of the marks achieved. Figure 6-16 summarises the grade steps achieved by all teams. 5 teams achieved one level and 20 teams achieved two levels out of a total of 28 teams. In total, 89% of the teams achieved at least 70% of the total points. 11% of the teams only passed the project without receiving any grade steps. Compared to the previous year's project, the percentage of successful to very successful teams increased by 49%. [5]

The calculation of improvement has the following limitations: If a project is repeated, it can be assumed that the material provided contains fewer errors and that the presentation and explanation of the approaches to be used becomes more understandable for the students. The comparison of the results of both projects is still limited because the project in 2021 uses an adapted approach in several steps.

[5] Notably, the scoring was not adjusted, so the overall result was better than the previous year's project.

	Very good results: at least 90% of the points		Good results: at least 70% of the points		Else: at least 50% of the points	
	Teams	%	Teams	%	Teams	%
MBSE Project 2020	1/21	7	7/21	33	13/21	60
MBSE Project 2021	5/28	18	20/28	71	3/28	11
Improvement		**11**		**38**		**-49**

Figure 6-16: Summary of the evaluation of the results of all teams.

6.4.7 Lessons learned

Overall, all teams were able to identify and model use case scenarios and security-related damage scenarios. The teams were able to derive requirements from the models and select and apply countermeasures based on the security design patterns provided.

Compared to the student reports from the previous year's project, the workload for the MBSE project had decreased. In the previous year's project, the potential of risk analysis was not utilised. All intermediate results were treated equally in the following activities, although the risk analysis would have allowed prioritisation.

Concerning Activity 0: See evaluation of competence test in Section 6.4.3.

Regarding Activity 1: Virtual collaboration in the concept phase requires the simultaneous processing of work results by several team members. This is easily done for modelling and requirements definition using Microsoft PowerPoint and Excel. 3DE, the 3D environment for visualising use case and damage case scenarios, was not used in this MBSE project because it did not allow simultaneous editing by multiple team members. For this reason I had decided that 3DE should be extended for this purpose for the next evaluation iteration (cf. Section 6.5.1).

Concerning Activity 2: As in the previous year's project, this MBSE project involved the creation of models at the system architecture level in Microsoft PowerPoint.

About Activity 3: In the previous year's project, I had used the SAHARA approach for risk analysis. This approach used three parameters to determine the security level. In this MBSE project, I used the attack potential approach. This approach allowed a more accurate determination of the security level because it used five parameters.

I had found that determining the safety level in the risk analysis process was causing difficulties for the participants. The ASIL risk classification scheme used for this purpose used one parameter to determine how often a driving situation could occur. The determination of this parameter was mostly based on 'gut feeling' and personal experience. Therefore, I had decided to support this decision with statistical data in the future. In Section 5.3 I report on this approach.

Regarding Activities 4 and 5: See Section 6.4.4.

Regarding Activity 6: As the project had shown, documenting model-based requirements in Microsoft Excel is possible in principle. Unfortunately, as the students reported, the effort required to track changes between models and requirements for the conceptual phase is very high. In Section 6.6 I report on creating models and documenting requirements using a professional MBSE tool.

6.5 Evaluation 4: A - Improved/New approaches

6.5.1 Evaluation of using a 3D environment

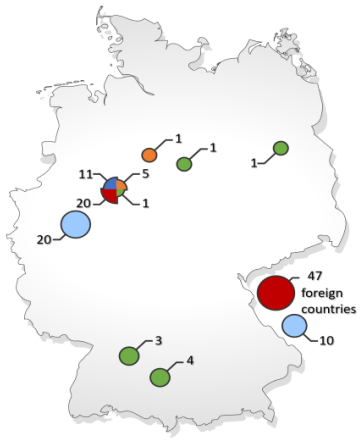
In this section I present the evaluation of the approach from Section 5.2. The approach and evaluation were presented in two papers [**JKK20**; **JSK+22**].

Most of the evaluation took place at the time of the COVID-19 pandemic. This had a major impact on the implementation of 3DE.

Instead of developing and using the tool for use in face-to-face workshops as planned, the tool had to be developed for use in online workshops. There were different constraints as the participants were working in different locations, in the company or in home offices, on different networks and with different operating systems.

6.5.1.1 Characteristics of the workshops conducted

In the following, I describe how and in what form 3DE and its method were evaluated. Over a period of 3 years, several online workshops were conducted with different groups of participants, on the basis of which the tool and the method were continuously improved. Participants came from different locations, countries and continents. Figure 6-17 shows a characterisation of the different workshops. Essentially, the workshops in the context of teaching at the University of Paderborn served to prepare for the application of the approach with subject matter experts. In addition to regular discussions on the progress of the approach with partners from industry and research, the approach was presented twice a year to a German car manufacturer in order to increase the practicability of the approach.



Geographical distribution of all 118 online-workshop participants

		Year	No of workshops	No of participants (average)	Identified use case scenarios (average)	Identified damage case scenarios (average)	Average workshop duration (h)	Workshop time per category (h)	Identified use cases or damage case scenarios /h	Design thinking steps considered	Design formalization steps considered
Teaching	Development team	1-3	10	4	1	1	2	80	1	yes	yes
	Summer school	2	1	30	14	14	8	240	3,5	yes	yes
	MBSE project	2	1	67	110	120	20	1340	11,5	yes	yes
Industry & research	Product development experts	3	1	7	1	1	1	7	2	yey	no*
	Automotive engineering experts	3	1	10	2	2	2	20	2	yes	no*

*This was done afterwards by me

In total, the tool and the method were applied in 14 workshops. The result is 266 modelled use case and damage scenarios and associated derived models. The approach could be tested in a total of 10.5 person-months (1 PM = 8h*20 days).

Figure 6-17: Characteristics of the workshops conducted.

A total of 14 online workshops were held during this period to improve the approach. In addition, there were numerous coordination meetings within the 6 supervised theses (cf. Section 5.2), which served to improve the tool in terms of usability and robustness. The basic version of the method was developed during the preparation of the first workshops.

6.5.1.2 Derivation of required features for 3DE

Figure 6-18 shows a list of the different development versions. It shows which features were available in which of the workshops presented in Figure 6-17. It also shows which development version fulfils which of the requirements defined in Section 5.2.2 and to what extent.

0.1	Proof of concept
1.0	New feature: Object builder
1.5	Experimental feature: VR support
1.7	Experimental feature: Local network collaboration
2.0	New feature: Web-App

Which requirements were satisfied by which version of 3DE?

Satisfied Partially s. Not satisfied

	3DE version:				
Considered requirements	**0.1**	**1.0**	**1.5**	**1.7**	**2.0**
R1: Suitable for cooperation in the concept phase					
R2: Enables synchronous collaboration					
R3: Enables the representation of structural relationships					
R4: Enables the representation of behaviour					
R5: Supports situational cognition					
R6: Provides low technical barrier to entry					
R7: Reduces effort for further use of the results					

Which version of 3DE was used in which workshops?

Satisfied Not satisfied

	3DE version:				
Applied in following workshops	**0.1**	**1.0**	**1.5**	**1.7**	**2.0**
Development team (Year 1 - 3)					
Summer school (Year 2)					
MBSE project (Year 2)					
Product development experts (Year 3)					
Automotive engineering experts (Year 3)					

Figure 6-18: Derivation of necessary and experimental features to improve 3DE.

The starting point for the tool was the unevaluated proof-of-concept implementation [**JKK20**]. This version was not suitable for use in online workshops. During the first workshops with the development team, it was found that certain 3D objects were not available (e.g. not free, complicated conversion required). For this reason, a plug-in [**Sch20**] was developed to create missing 3D objects in a simplified form using voxel blocks in just a few steps. This development phase served as the basis for a full-day online workshop with 30 master students from the field of computer science & engineering (cf. Section 6.2). Due to the large number of participants, the development of the results took place in groups. One group member at a time used the tool, while the other group members provided information for modelling.

In order to improve immersion, an extension for Occulus Quest [**Kor20**] was experimentally developed for use with VR hardware. The immersion effect was impressive, as one could virtually move around the 3D environment from both first person and bird's eye perspectives. As online workshop participants do not usually have such hardware, this extension was not developed further. However, this extension would be an added value for local workshops, where the VR hardware would be brought by the workshop organisers.

Enabling the simultaneous elaboration of use cases and damage scenarios within 3DE was done in two stages. In Stage 1, the focus was on developing functions for simultaneous operation of the 3D environment by multiple participants (simultaneous placement/moving of

objects, blocking of simultaneous editing, etc.). This was tested in development workshops in a defined network [**Sch21b**].

In Stage 2, the focus was on network-independent (and thus primarily location-independent) simultaneous use of the tool. This enabled online workshops with simultaneous editing by several participants in the same 3D environment. To ensure a broad and platform-independent use of 3DE, it was further developed as an application in the web browser. This version could be accessed via a web link for the fastest possible start. To further reduce technical barriers, WebGL [**KHR22-ol**] was chosen, which could display 3D graphics in the browser without the need to install additional plug-ins. This level of development was a prerequisite for the two final expert workshops (cf. Section 6.5.1.3).

The simultaneous elaboration of use cases and damage scenarios within 3DE was done in two stages. In Stage 1, the focus was on developing functions for simultaneous operation of the 3D environment by multiple participants (simultaneous placement/moving of objects, blocking of simultaneous editing, etc.). I tested this in development workshops in a defined network. In Stage 2, the focus was on network-independent (and thus primarily location-independent) simultaneous use of the tool. This led to the development of a cloud solution using Photon Unity Networking 2 [**PUN22-ol**]. In order to ensure a broad and platform-independent use of the tool, I decided to further develop it as an application in the web browser that can be invoked via a web link. WebGL [**KHR22-ol**] was used for this, as it allowed 3D graphics to be displayed in the browser without technical barriers. This stage of development provided the conditions for the two final expert workshops to be held virtually, independent of location, network and operating system.

6.5.1.3 Expert feedback

In general, the approach was well received by the 17 experts (cf. Figure 6-19, W1-W4). I would like to highlight W4 *The approach can be used in industry. Generated SysML models can be refined, and the visualisation can be inserted as a screenshot into any requirements engineering/architecture design tool.* One can easily fix C1 by limiting the use of very detailed 3D models. C1 also depends on the available bandwidth of each workshop participant. C2 was mentioned three times. This is because only verbal facilitation was used for each step. A solution would be to use a workshop canvas in a 2D workshop tool, where the steps and the artefacts required are described in detail. In addition, each step would be supported by a simple example.

Change	Worked
C1: Placement of 3D objects is a bit slow.	W1: Intuitive step by step approach.
C2: The task should be more precise (3x).	W2: Structured approach, also without prevous knowledge.
C3: I don't know if the tools we use are appropriate for the complexity of the task. Maybe we should do this in person, or reduce the complexity.	W3: The tool was very visual and quite easy to handle.
	W4: The approach can be used in industry. Generated SysML models can be refined and the visualization can be inserted as a screenshot into any requirements engineering/architecture design tool.
Questions	**Ideas**
Q1: What would be the next step and did we interprete it the right way or not?	I1: Consideration of factors such as fog, rain, reflections, etc. in the tool.
	I2: Specifying the application and damage cases by designing the 3D environment.
	I3: Precision of object relationships and sequences through parameters such as speed, weather, etc.

Figure 6-19: Feedback from the 17 participants of the two expert workshops.

Alternatively, 3DE could guide users step-by-step through the creation of 3D scenarios. However, I see the danger of limiting creative elaboration, which often does not follow a strict flow. C3 is partly related to C2. The online workshops could also be conducted as face-to-face workshops. As the workshop participants came from all over Germany, most of them would have to travel a long way to the workshop venue and back. This would add two travel days to the total time spent per workshop. This contradicts the limited time that workshop participants have in their daily project work. I1-I3 are suggestions for extending 3DE. I consider all extensions to be useful for future work, although in the case of I2 the cost/benefit ratio in 3D modelling needs particular consideration.

6.5.2 Evaluation of the use of statistical data

In this section I present the evaluation of the approach from Section 5.3. The approach and evaluation were presented in [**JKA+22**].

In the context of my industry contacts with a car manufacturer, two car suppliers and in the context of an automotive conference [**ESC21-ol**], I discussed the use of statistics in the conceptual phase of risk assessment with several experts in the field.

I found from the discussions that risk assessment in the concept phase is done on the basis of personal experience and "gut feeling". In addition, the experts mentioned that only certain people have access to and knowledge of statistically validated data in later stages of the development process. In general, all experts were interested in easy-to-use, statistically validated risk assessment tools for use in the concept phase.

In preparation for the evaluation with a total of 17 experts in two workshops (cf. Figure 6-20), I first conducted two workshops with students to find optimisation potential in the

Category	Description	No of participants	No of identified damage scenarios	Workshop duration in hours	Workshop time per category in person hours
Teaching	Student workshop 1	3	1	2	6
	Student workshop 2	3	1	2	6
Industry & research	Product development experts	7	1	1	7
	Automotive engineering experts	10	2	2	20

Figure 6-20: Characterization of the conducted workshops.

Change	Worked
C1: The task should be more precise (3x).	W1: Intuitive step by step approach.
C2: I don't know if the tools we use are appropriate for the complexity of the task. Maybe we should do this in person, or reduce the complexity.	W2: Structured approach, also without prevous knowledge.
	W3: Assessing the questions from the statistics table was comparatively easy once we kind of agreed on how to interprete the given task.
Questions	**Ideas**
Q1: What would be the next step and did we interprete it the right way or not?	I1: Start with statistics, then modeling of damage scenarios.
Q2; How specific should the damage scenario be selected.	I2: Would it be useful to use the statistical data to construct scenarios to ensure that realistic (and probable) scenarios are used (instead of those, that are only from personal experience and thus may be not as important as it seems).
Q3: Do we investigate alternative ways to derive risks? Instead of using individual scenarios with high details, we could use more general problematic situations?	I3: Can be easily implemented in Excel file with selection of ASIL parameters instead of looking up in tables.

Figure 6-21: Feedback from the professional experts on the approach and characterisation of the workshops conducted.

approach in practical use. Then I conducted a workshop with 7 product development experts. Finally, I conducted a workshop with 10 experts from the field of automotive engineering. My role was to prepare and moderate the workshops.

All workshops were conducted online using Microsoft Teams and an online collaboration tool. I have listed the feedback from the workshops with the experts in Figure 6-21. Based on this feedback, I derived the next steps.

In principle, the approach was well received (cf. 6-21, W1-W3). It was mentioned several times that the terms of reference were unclear (C1). This was due to the fact that, at the time of the workshop, I did not fully visually support the relationships and processes shown in Figure 5-7. The Federal Statistical Office did not provide data for all severity levels in some cases (cf. Figures 5-8 - 5-10). My suggestion to use the closest severity level instead led to some uncertainty among participants.

Due to the COVID-19 pandemic at that time, C2 was not feasible at that time. Q1 could be resolved by review by other experts in the field. (Q2) Further work could be done in the

future to determine what guidelines would be helpful in determining the appropriate level of granularity. Q3, I1 and I2 suggest a different order of processing steps. In principle, one could alternatively start with critically evaluated scenarios and then model or discuss a specific scenario. However, I think that in this case the risk assessment should be checked again against the concrete scenario. (I3) In the future, a web application containing the elements of the approach could facilitate the work in online workshops.

6.5.3 Evaluation of using a tool for model transformation

In this section I present the evaluation of the approach from Section 5.4. The approach was developed in the context of five industrial projects with a German premium car manufacturer [**OEM19b**; **OEM20**; **OEM21a**; **OEM22a**; **OEM22c**]. Over a period of three years a tool was developed that extracts ECML models from Microsoft Visio, maps them to SysML and generates SysML models for the MBSE tool Cameo Systems Modeler.

First, I explain the process of identifying customer requirements. I then justify the decision to redevelop the resulting prototype. Then I mention the ECML system architectures used for the evaluation. Finally, I explain the evaluation of the iteratively developed prototype.

The prototype was developed using the following procedure: (1) First, the customer's user requirements were identified. (2) These were then implemented by the development team. Implementation and white-box testing took place in a development environment. (3) Based on the white-box testing, bug fixes were performed by the development team. (4) The prototype binaries were then handed over to the customer's testing team for black-box testing in the target environment. (5) Based on the bug report from the test team, the bug was fixed by the development team. (6) The prototype results were reported to the customer. Based on the customer's feedback, the process was repeated, if necessary, for new or modified customer requirements.

For Cameo Systems Modeler (CMS) there was a plug-in that could generate SysML models from model information in a text file. Instead of using the existing plug-in, a new development was preferred for the following reasons: The plugin only supported 7-bit

ECML models	
M1	Training example: Interchangeable license plate system
M2	Training example: Electric tailgate - Simple version
M3	Training example: Electric tailboard - Detailed version
M4	Annonymized model from an ongoing project

Model characteristic	**M1**	**M2**	**M3**	**M4**
Hierarchized	No	No	Yes	Yes
ECML blocks	9	11	16	14
ECML interactions	16	29	35	35
ECML ports	32	58	70	70
Effect types	32	58	70	70
Text descriptions	25	40	51	49
Sum of model elements to be exported	**114**	**196**	**242**	**238**

Figure 6-22: Characterisation of the ECML models provided by the customer.

ASCII encoding. This caused, for example, German special characters to be displayed incorrectly in generated SysML models. This required manual post-processing of text descriptions in SysML models. The plugin did not support user-defined stereotypes. As a result, ECML blocks and interactions could not be mapped to SysML as described in Section 5.4.2. In addition, further development of the plugin for future CMS releases has been discontinued.

In Figure 6-22 I present an overview of the ECML models provided by the customer. The first three models M1-M3 are training examples. They were used by the car manufacturer to train internal and external staff in the use of ECML. In terms of model complexity, these models were at the level of real project deliverables.

M4 was a model from an ongoing project. In order to ensure the functionality of the tool to be developed with real models, I needed such models. Through discussions with the car manufacturer, we came up with the idea that the models could be provided in an anonymised form. In an anonymised model, the model names were replaced by a meaningless string. In this case it was the string “abc”. The anonymisation had no effect on the functionality of the tool to be developed. M4 was developed within a few months at the car manufacturer in four versions, which were made available to me in anonymised form.

The M1 model had a low level of model complexity. The model did not consist of multi-layered model elements and had few blocks and interactions. M2 had medium model complexity because it had considerably more interactions between blocks than M1. M3 and M4 were complex models that contained additional blocks that were layered. Based on my experience from industrial projects, M1 and M2 were equivalent to more advanced results in the conceptual phase, while M3 and M4 had the complexity and scope of completed results.

Figure 6-23 shows a comparison of the developed prototypes with the specified requirements. Based on the initial requirements, the first prototype was created and evaluated using M1. Based on these results, new requirements for the next version of the prototype were evaluated with additional models, etc. At the time of documenting this work, the prototype was to be used by the car manufacturer to perform a model transformation for 50 models.

R1	Extraction of ECML blocks
R2	Extraction of ECML ports
R3	Extraction and mapping of effect types to ECML ports
R4	Extraction of explicitly linked interactions
R5	Extraction of hierarchized ECML blocks
R6	Extraction of the flow direction of ECML ports
R7	Extraction of implicitly linked interactions
R8	Plausibility check of ECML models

To what extent does the iteratively developed prototype satisfy the requirements?

Satisfied | Partially s. | Not satisfied

Iteration	Evaluated on				Requirements							
					R1	R2	R3	R4	R5	R6	R7	R8
V.2020a	M1											
V.2020b	M1											
V.2021a	M1	M2										
V.2021b	M1	M2										
V.2022a	M1	M2	M3	M4 (V.1)								
V.2022b	M1	M2	M3	M4 (V.2)								
V.2022c	M1	M2	M3	M4 (V.3)								
V.2022d	M1	M2	M3	M4 (V.4)								

Figure 6-23: Comparison of the developed prototype versions with the specified requirements.

6.5.4 Lessons learned

With the help of 3DE, online workshops were held to create and discuss damage scenarios. This was used to assess the impact of the damage scenarios. A damage scenario usually involved a specific road segment. I found that participants spent a relatively large amount of time modelling a road segment using placeable road segments.

One way of reducing this effort was to use a large 3D model of a small town [**Win18-ol**]. Due to the size of the model, this solution was not suitable for use in online workshops as the loading times were too long and online participants would need a relatively powerful computer to use it. In addition, the road network of the small town was only specific to the USA.

Accident data could be used to identify dangerous road sections where a modelled damage scenario would have a critical impact. The German Federal Statistical Office's accident atlas [**Sta22-ol**] would be one way of doing this. This contained over 10 million traffic accidents recorded by the police on German roads for the period 2017-2021. The accidents were recorded with GPS accuracy. The data included information on which accidents had occurred at which locations, as well as their severity and frequency.

(2) To reduce the search effort for critical road sections, the data set of the Federal Statistical Office was integrated into 3DE together with the Google Maps extension as part of the master thesis I supervised. This enabled the data-driven modelling of damage scenarios in 3DE for critical road sections in Germany (cf. Section A.1 for more details).

The development of the model transformation tool consisted of several projects that built on each other. In each project I had several consultation meetings with the vehicle

manufacturer. These meetings and regular demonstrations of the development results led to the success of the projects. The continuous coordination enabled the prototype development work to be regularly adapted to the evolving requirements of the car manufacturer.

6.6 Evaluation 4: B - Evaluation with subject matter experts

6.6.1 Overview

In this section I present the evaluation of my final approach. In doing so, I follow the process model from Section 5.9. I also show how I integrate the approaches I developed (cf. Sections 5.2 - 5.8) into the overall approach. The main part of my final approach I realised with the MBSE tool Cameo Systems Modeler. In doing so, I use the SysML profile I created (cf. Section 6.6.8), which facilitates the creation of ISO/SAE 21434 work products for the concept phase.

I use traffic sign recognition in vehicles as a continuous application example (cf. Section 6.6.2).

For clarity, this section shows only a part of the full application example, which is necessary to explain my approach. The full application example is shown in the appendix in the Section B.1.

I also report on my real-life test (cf. Section 6.6.9). The purpose of the test was to obtain an application example that is as realistic as possible. Finally, I report on the presentation of my final approach in the context of three workshops with experts from industry and research in the automotive sector (cf. Section 6.6.10).

		Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
Concept phase	[WP-09-01] Item definition	x			x	
	[WP-09-02] TARA		x	x		x
TARA methods	[WP-15-01] Damage scenarios		x			
	[WP-15-02] Assets with cybersecurity properties		x			
	[WP-15-03] Threat scenarios			x		
	[WP-15-04] Impact ratings with associated impact categories		x			
	[WP-15-05] Attack paths					x
	[WP-15-06] Attack feasibility ratings					x
	[WP-15-07] Risk values					x
	[WP-15-08] Risk treatment decisions					x
	[WP-09-03] Cybersecurity goals					x
	[WP-09-04] Cybersecurity claims					x
	[WP-09-05] Verification report for cybersecurity goals					x
	[WP-09-06] Cybersecurity concept					x
	[WP-09-07] Verification report of cybersecurity concept					x

Figure 6-24: Overview of my approach for creating the 15 work products of the concept phase of ISO/SAE 21434.

6.6.2 Application example - Intelligent Speed Assistance

My application example is the intelligent speed assistant. It warns the driver if the speed limit is exceeded. The speed assistant gets its speed limits from traffic sign recognition (and/or data from navigation services). Speed assistants will be mandatory for newly registered vehicles in the EU from July 2024 [**TUEV22b-ol**] and are realised by several vehicle components interacting with each other.

I also assume the presence of an automatic speed limiter. This adjusts the speed depending on the traffic signs detected (cf. Polestar 2 [**Pol22-ol**], Tesla Model 3 [**Tes23-ol**]).

A possible threat could come from a Digital Signage System (DSS). A DSS is a remotely controlled digital display used as an advertising medium. Such systems are often located on roadsides. I assume that an attacker could manipulate a DSS to display a traffic sign and mislead the intelligent speed assistant, causing a dangerous situation.

6.6.3 Phase 1: System analysis at environment level

6.6.3.1 [WP-09-01] Item

This phase looks at the item from a black box perspective, the big picture. This phase (together with Phase 4) deals with the work product [WP-09-01]. According to my approach, vehicle functions are first defined in the form of use cases. For this I use SysML use case diagrams. For these use cases, environment systems or environment elements interacting with the element are identified. For traceability purposes, links are made to elements of the use case diagrams.

6.6.3.2 Requirements

I use SysML requirements in tabular form. In my case, seven black box requirements were derived for the use case *Traffic Sign Recognition*, taking into account the environment elements *Traffic Sign* and *Driver* (cf. Figure 6-25 and 6-26).

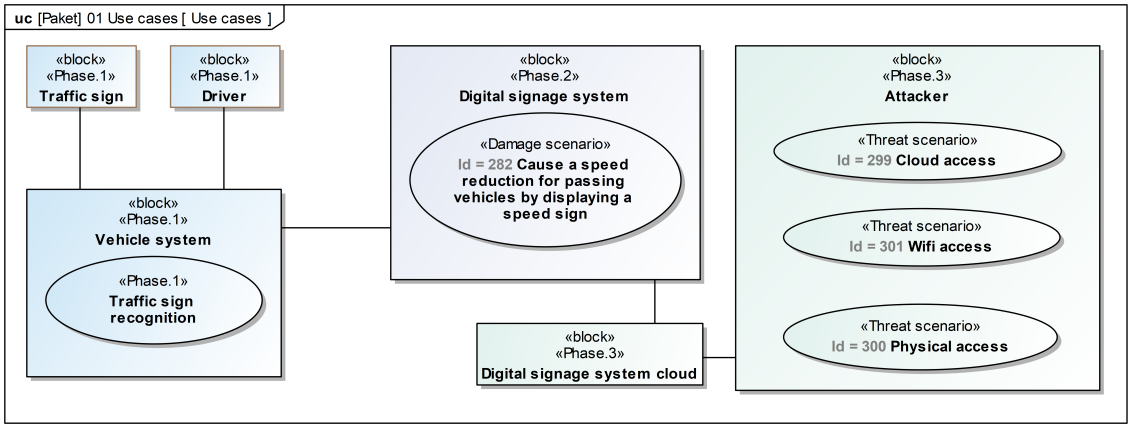


Figure 6-25: Overview of use cases, damage scenarios and threat scenarios at the system environment level (related to [WP-09-01], [WP-15-01], [WP-15-03]).

#	△ Name	Requirement description
1	122 Active traffic sign assistant	The system shall actively support the driver in recognizing traffic signs and assist the driver in the prevention of damage.
2	122.1 Traffic sign recognition	The system shall support the driver in traffic sign recognition.
3	122.2 Reaction to recognized traffic signs	The system shall warn the driver and actively assist in the prevention of damage.
4	122.2.1 Driver warning	The system must be able to warn the driver.
5	122.2.1.1 Speed sign warning	The system shall warn the driver when the detected maximum speed is exceeded.
6	122.2.2 Driver assistance	The driver shall be actively supported by the system in reacting to recognized traffic signs.
7	122.2.2.1 Speed assistance	The system shall make it possible to automatically reduce the speed of the vehicle to the detected maximum speed.

Figure 6-26: Derived Requirements (related to [WP-09-01]) for the use case Traffic Sign Recognition.

6.6.4 Phase 2: Impact analysis at environment level

6.6.4.1 [WP-15-01] Damage scenarios

ISO/SAE 21434 requires the identification of damage scenarios [WP-15-01]. A damage scenario is an adverse effect on a vehicle or a vehicle function with consequences for a road user.

The approach in Section 5.2 can be used to identify damage scenarios. Using a 3D environment, damage scenarios such as the one shown in Figure 6-27 can be modelled and discussed. The scenario shows an attacker who has manipulated a DSS to display a speed sign. This affects the traffic sign recognition of the silver vehicle, resulting in a speed reduction and a possible collision with the approaching yellow vehicle.

Figure 6-27: Manipulation of a digital signage system, by displaying a traffic sign, and possible collision as a result.

SysML use case diagrams are used to collect the names of the damage scenarios. SysML requirements tables are then used to refine the damage scenario description. A link from the damage scenario description to a picture of the modelled scenario improves the understanding of the damage scenario (cf. Figure 6-28).

#	Name	Damage scenario description	Traced To
1	282 Cause a speed reduction for passing vehicles by displaying a speed sign	The digital signage system is located (DSS) on the right side of a road in the city. The maximum permitted speed is 70 km/h. Vehic A and vehicle B are driving at 70 km/h on the road. Vehicle B has a small distance to vehicle A. Vehicle A has sign recognition and speed reduction turned on. An attacker hacks the DSS and uploads a 20 km/h sign to the DSS. As a result, vehicle A abruptly reduces its speed to 20 km/h. Vehicle B crashes into vehicle A.	Visualization - Damage scenario - Traffic sign recognition

Figure 6-28: Description of a damage scenario as part of [WP-15-01].

6.6.4.2 [WP-15-02] Assets

In [WP-15-02], ISO/SAE 21434 requires the identification of assets and their cybersecurity properties whose compromise results in a damage scenario. An asset is an object (e.g. a component) that has or contributes to value. An asset has one or more cybersecurity properties that, if compromised, can lead to one or more damage scenarios. A cybersecurity property is an attribute that may be worth protecting. Such attributes are confidentiality, integrity and/or availability.

The affected assets are identified using the damage scenarios listed in a table. The violated cybersecurity properties are then derived. In Figure 6-29 I show the damage scenario *Cause a speed reduction for passing vehicles by displaying a speed sign.*

#	Name	Asset	Cybersecurity property
1	282 Cause a speed reduction for passing vehicles by displaying a speed sign	Multi purpose camera	Integrity

Figure 6-29: Determination of affected assets and compromised cybersecurity properties of a damage scenario, as part of [WP-15-02].

By displaying a traffic sign on a DSS, the detection of the asset *Multi Purpose Camera* is tricked. This violates the cybersecurity property *Integrity*[6].

6.6.4.3 [WP-15-04] Impact rating

As part of [WP-15-04], the damage scenarios have to be rated according to the potential negative consequences for road users in the categories of *safety, financial, operational and privacy (S, F, O, P)*. The assessment is made in four levels: *severe, major, moderate or negligible*. With regard to safety, reference is made to the ASIL classification scheme of ISO 26262. The additional parameters *exposure* and *controllability* are used together with a mapping table to determine the safety impact.

In Section 5.3 I presented an approach to determine the safety impact of a damage scenario using statistical data. This was done by combining the ASIL classification scheme with aggregated data from the Federal Statistical Office. Figure 6-30 shows which questions and tables were relevant for the determination of the safety impact. To determine the exposure value, I examined the table entries and checked for consistency with the damage scenario under consideration. Basically, I assumed that the worst-case damage scenario would result in severe injuries (severity S2) and therefore not fatal injuries (S3). In order to establish compatibility between the ISO 26262 ASIL classification scheme and the ISO/SAE 21434 specifications, a mapping has to be performed. To do this, the severity levels S0,S1,S2,S3 must be mapped to the levels negligible, moderate, major and severe.

---

[6] For better understanding, the *Multi-purpose camera* has been used in this phase. This is in anticipation of the identification of the system components required to realise the use cases in Phase 4.

I also assumed that the time window for the driver's reaction (controllability) was in the medium range (C2, Normal). In particular, it is neither an unavoidable situation (C3, Difficult) nor an easily avoidable situation (C1, Easy). On the basis of these inputs, the values in Figure 6-30 are determined. In most cases the result will be an ASIL value of B. I take this information and the ASIL value for the safety part of the impact rating in Figure 6-31. I assume the same level of severity for the financial and operational impact. As no significant personal data is used in this case, I set the privacy impact to negligible.

Question-based selection of relevant tables			Relevance	Relevant entry from the table	Exposure	Severity	Controllability	ASIL
What non-driver-related causes can play a role as the cause of the accident?	Are obstacles, weather or road conditions relevant?	3	Yes	Rain on street	E4	S2	C2	B
What other aspects can play a role as the cause of the accident?	Do I know the road users who suffer injury?	6		Persons in the vehicle	E4	S2	C2	B
	Is the type of road user of the main cause of the accident known?	7		Persons in the vehicle	E4	S2	C2	B
	Are types of road users known who have suffered damage as a result of the accident?	8		Persons in the vehicle	E4	S2	C2	B
	Is the damage location or damage type relevant?	10		Outside buid-up areas, Personal injury	E3	S2	C2	A
	Is it only property damage or personal injury as well?	11		Personal injury	E3	S2	C2	A

Figure 6-30: Determination of the safety impact with the help of statistical data.

#	Name	Safety impact - Exposure	Safety impact - Controllability	Safety impact - ASIL	Safety impact - Severity	Financial impact	Operational impact	Privacy impact
1	282 Cause a speed reduction for passing vehicles by displaying a speed sign	Medium	Normal	B	Major	Major	Major	Negligible

Figure 6-31: Impact ratings for safety, financial, operational and privacy as part of [WP-15-03].

6.6.5 Phase 3: Security analysis at environment level

6.6.5.1 [WP-15-03] Threat scenarios

In Phase 3, based on the system and impact analysis, threat scenarios are identified and documented using SysML Use Cases (cf. Figure 6-25). This is followed by a detailed description using SysML Requirements [WP-15-03] (cf. Figure 6-32). A threat scenario is a possible cause for compromising the cybersecurity properties of one or more assets in order to achieve a damage scenario. In our use case, an attacker could manipulate the

DSS to trick the *Traffic Sign Recognition* of a vehicle using the following three approaches: *physical access, wifi access, cloud access*.

In the following I will focus on a possible attack via *cloud access*. A DSS can be managed remotely. This requires credentials. In Figure 6-32 I describe a possible threat scenario. UN R155 categorises and lists a total of 77 threats. The regulation requires evidence for the CSMS that these threats have been checked. In my approach, the threats are included in the form of SysML requirements. I realise the proof in Phase 3 by linking the threat scenarios directly to the corresponding UN R155 threats.

#	Name	Threat scenario description	Related threats from UN R155
1	299 Cloud access	Attack on the cloud system via brute-force attack on the login page	222.3.1 (28.1) Software bugs. The presence of software bugs can be a basis for potential exploitable vulnerabilities. This is particularly true if software has not been tested to verify that known bad code/bugs is not present and reduce the risk of unknown bad code/bugs being present

Figure 6-32: Description of a threat scenario in the context of [WP-15-03] and referencing to UN R155 threats.

6.6.6 Phase 4: Analysis at system level

6.6.6.1 [WP-09-01] Item

In this phase, the item is considered from a white box perspective. The focus is on identifying the components and component relationships required to realise the use cases under consideration. This phase (together with Phase 1) deals with the work product [WP-09-01].

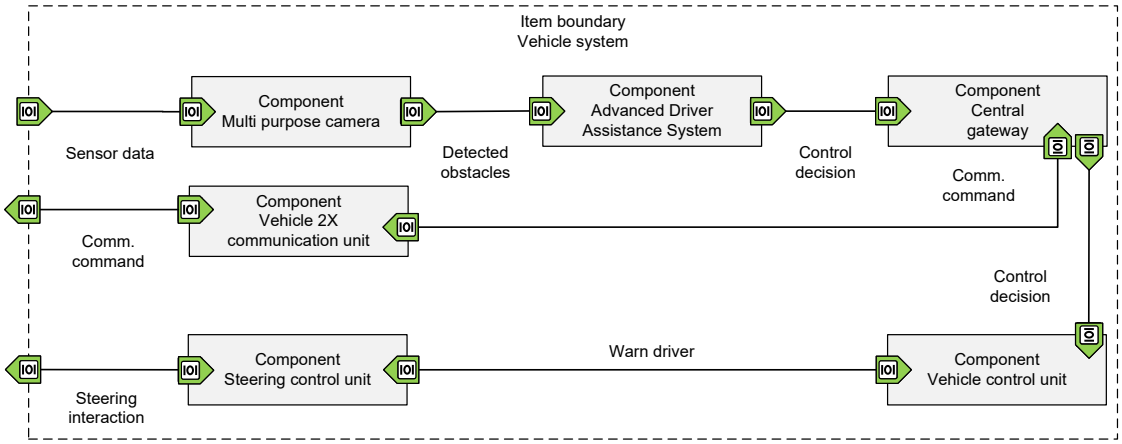


Figure 6-33: Identification of necessary components and component relationships in the context of the item under consideration.

Based on my project experience with a German premium car manufacturer, I assume that the item was developed in a workshop with an interdisciplinary stakeholder team. I assume that the stakeholders are mainly managers with holistic knowledge in several development areas. At the same time, I assume that these stakeholders do not have in-depth expertise

in modelling SysML models. I also assume that the models have been created using the Effect Chain Modelling Language (ECML) (cf. Section 5.4). ECML models consist of simple language constructs, are used in practice and are easy to create using Microsoft Visio. Figure 6-33 shows an ECML model that could be the result of such a workshop. The model refines the model shown in Figure 6-25 with components and relationships to realise the use cases considered in Phase 1.

Use cases can be implemented using different components and component relationships. In order to create an architecture that is as realistic as possible, the components and component relationships from the 5.5 Section have been used. These are the result of an examination of 69 product descriptions of safety-related Bosch components (cf. Section 5.5). To transform ECML models into SysML models, I use the mapping between ECML and SysML presented in Section 5.4 and the tool developed for this purpose. The result of the model transformation is the SysML Internal Block Diagram (IBD) shown in Figure 6-34. In particular, the model contains the information from the ECML model about the defined relationship types. In this case, all component relationships are of type Intentional. The information about the ECML effect types is stored in an interface diagram (cf. Figure 6-35).

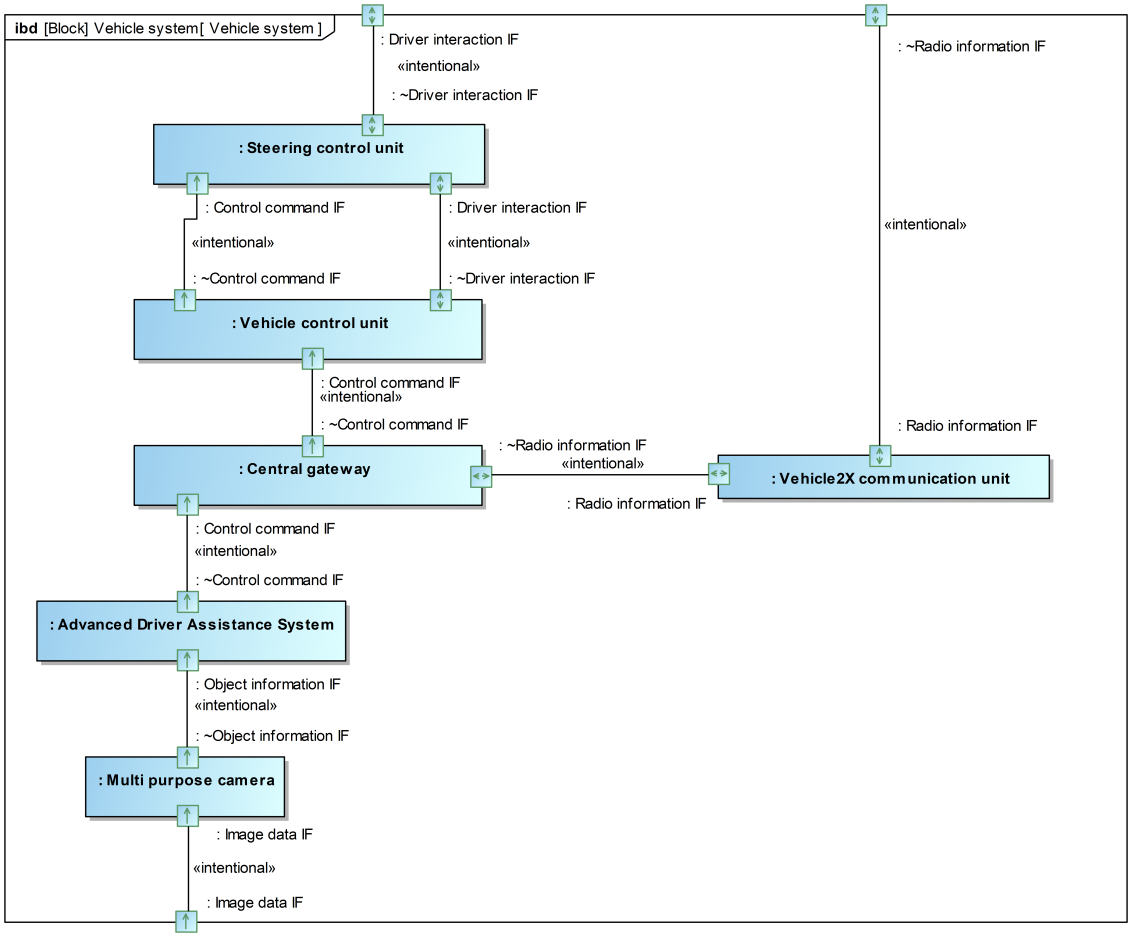


Figure 6-34: Identification of necessary components and component relationships for the realisation of the considered use case as part of [WP-09-01].

Component relationships are implemented using ports. Proxy ports can be used to model complex component relationships between two ports. Proxy ports are realised by an interface block that can be used to specify multiple flow properties. This significantly reduces the number of ports to be modelled in an IBD. The increased clarity and reuse of interface specifications reduces the potential for errors. The proxy port *Driver Interaction IF* contains the flow properties *Driver Warning* and *Driver Reaction*. The *Driver* interacts with the *Controller* via these flow properties.

Figure 6-35 shows the interface blocks used for the model in Figure 6-34. This allows the ports to be specified in more detail. The figure also shows the effect types defined in the ECML model. In this case, the Information/Software effect type is used.

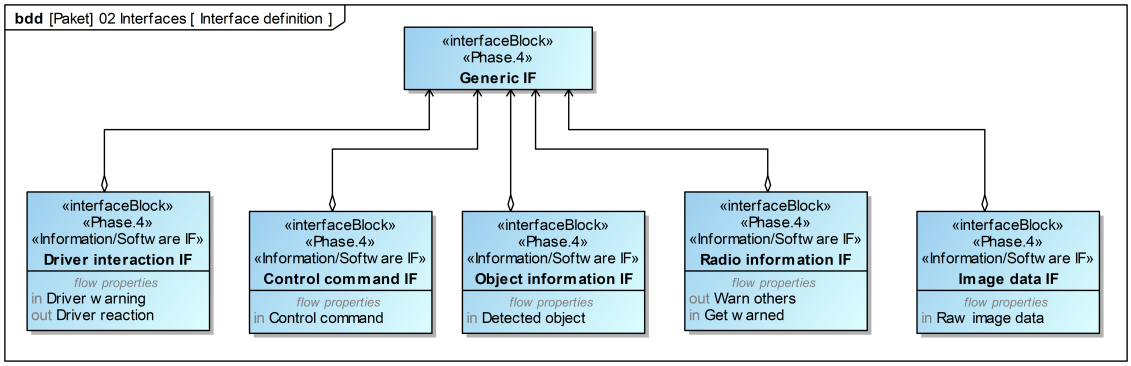


Figure 6-35: Interface definition.

Driver warning and *Driver reaction* are directional flow properties. In my case, *Driver warning* is specified in the incoming direction and *Driver reaction* is specified in the outgoing direction. If two components A and B communicate via the same proxy port, the flow directions of a proxy port must be inverted for one component so that, for example, outgoing signals can be treated as incoming signals on the other component. Since manual inversion is time-consuming and error-prone, I use conjugated (proxy) ports. Here the described behaviour is automatically included. Such ports are marked with the prefix $\sim$ according to the SysML specification.

In my approach the components and the general component relations are modelled top-down. This is followed by the detailed specification of flow properties within interface blocks. Finally, the conjugated ports are specified.

6.6.6.2 Requirements

Based on the IBD, the requirements from Phase 1 are refined (cf. Figure 6-36). This reveals ambiguities in the modelled IBD and increases the understanding of the IBD for people not involved in modelling.

#	Name	Requirement description.	Traced To
1	292 Sign recognition	The vehicle must dedect traffic signs with the help of a camera sensor.	Use cases Vehicle system
2	292.1 Processing of object information	The camera sensor must send information about detected traffic signs to the Advanced Driver Assistance System so that the Advanced Driver Assistance System can derive a decision.	Multi purpose camera Advanced Driver Assistance System
3	292.1.1 Forwarding ADAS decision	The decision of the ADAS must be forwarded via the Central Gateway, the Vehicle Control System up to the Steering Control System.	Central gateway Vehicle control unit Steering control unit
4	292.2 Driver warning through Steering Control	If the Steering Control receives a message from the ADAS about the Vehicle Control, the driver must be warned.	Steering control unit Vehicle control unit Advanced Driver Assistance System
5	292.2.1 Reaction to driver warning	If the driver confirms the warning or does not react, the Steering Control must inform the Vehicle Control. If not, it must not.	
6	292.3 Adjust speed	If the Steering Control informs the Vehicle Control that the driver has acknowledged or ignored the warning and thus has not discarded it, the Vehicle Control must adjust the speed.	Steering control unit Vehicle control unit

Figure 6-36: Refinement of the requirements from Phase 1 from a white box perspective (related to [WP-09-01]).

6.6.7 Phase 5: Security analysis at environment level

6.6.7.1 [WP-15-05] Attack paths

According to ISO/SAE 21434, the identified threat scenarios should be used as a starting point to derive attack paths. An attack path consists of a number of attack steps. The set of attack paths is defined by [WP-15-05].

Individual attack steps may appear in several attack paths. Furthermore, attack paths can be part of larger attack paths. Creating copies of attack steps and attack paths in different contexts has the potential for error, as changes must always be made in all copies. This makes it unnecessarily difficult to perform an impact analysis.

My approach is as follows: I model attack steps as separate and reusable elements to facilitate reuse. I aggregate multiple attack steps into attack paths. Since attack paths can overlap in individual attack steps, I use attack trees to provide a comprehensible representation. Attack trees can be extended to fault trees by the additional use of logic gates [**ESH25**]. Logic gates (e.g. AND gate, OR gate) can be used to model the cause-effect relationship between attack steps at different levels (cf. Figure 6-37). With the help of a transfer gate, represented by a triangle, a reference to a sub-tree can be established, so that the reuse of already modelled sub-trees is supported.

In my work, I use SysML requirement diagrams to model fault trees. This means that I do not need a separate tool. This ensures the traceability of already modelled elements from previous phases. Individual diagram elements (attack steps, logic gates, transfer

elements) correspond to SysML requirements and are distinguished from each other by stereotypes. For better representation, the symbols for logic gates and transfer elements are automatically adapted depending on the stereotype. The use of requirements also facilitates reuse in subsequent steps.

Figure 6-37 shows an attack tree. The attack tree addresses the Phase 3 damage case *Causing a speed reduction for passing vehicles by displaying a speed sign*. The attack tree has been modelled with the assumption that an attacker can achieve the damage scenario in at least one of three possible ways. Access via cloud, wifi or physical access. This was implemented using an OR gate. Each access path consists of its own subtree. In addition, it is necessary that the attacker has created the content to be displayed on the DSS (in this case, the 20 km/h speed sign) and has assigned this content to a time period. I have used an AND gate because the attack on the DSS requires the creation of content for the DSS and the scheduling of content for the DSS.

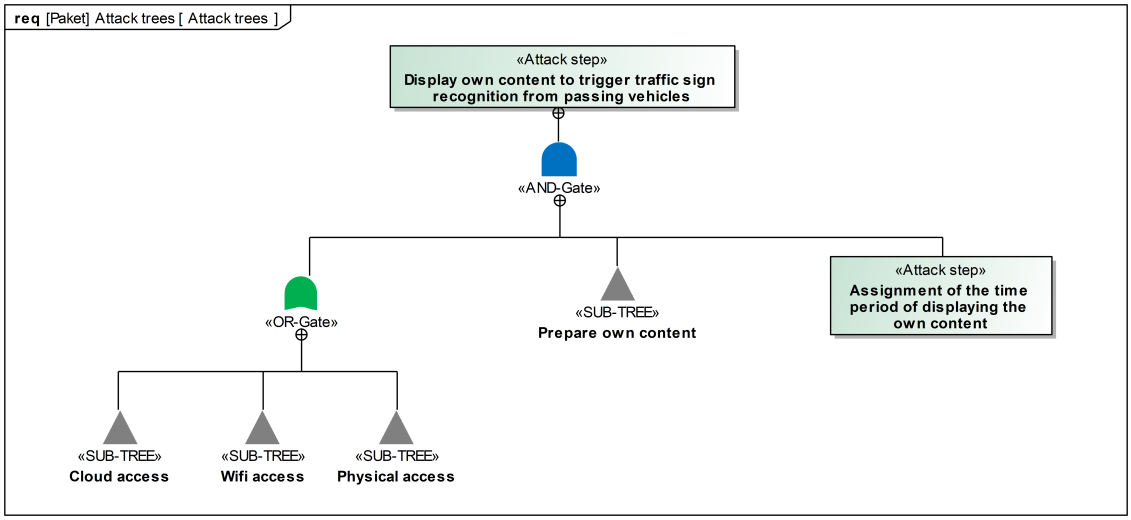


Figure 6-37: Overall attack tree, for tricking the object recognition of a vehicle by a digital signage system (related to [WP-15-05]).

6.6.7.2 [WP-15-06] Attack feasibility rating

According to ISO/SAE 21434, an attack feasibility evaluation [WP-15-06] shall be performed for identified attack paths. The rating scale to be used is *Very low, Low, Medium and High*. For *Very low*, the attack path can only be executed with a very high effort. With *High*, the attack path can be executed with very little effort.

ISO/SAE 21434 suggests the attack potential based approach as the first choice for the attack feasibility rating [**ISO21**]. This uses the factors of elapsed time, expertise, knowledge of the item or component, window of opportunity and equipment to determine an Attack Feasibility Level (AFL) using a mapping table. Each of these factors has several levels of expression.

In the approach, I implement the attack potential based approach as follows (cf. Figure

#	Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.1 Cloud access						Medium
2	163.1.1.1.2						Medium
3	163.1.1.1.2.1 Tool automated brute force access to the login website						Medium
4	163.1.1.1.2.1.1						Medium
5	293.3.1 Examine source code of login web page for remote access to login fields	Expert	Restricted	Specialized	<= 1 week	Unlimited	Medium
6	293.3.2 Search online for tools to attack login pages (brute force login)						High
7	293.3.2.4						High
8	293.3.2.4.1 Search online for tools to attack login pages (password list generator)	Proficient	Public	Specialized	<= 1 week	Unlimited	High
9	293.3.2.4.2 Generate password list, according to password specification for login page	Proficient	Public	Specialized	<= 1 week	Unlimited	High
10	293.3.2.4.3 Gather information about password specification of content management software for digital signage systems	Layman	Public	Standard	<= 1 week	Unlimited	High
11	163.1.1.1.2.2 Upload prepared own content to the cloud content management system	Layman	Public	Specialized	<= 1 day	Unlimited	High

Figure 6-38: Attack feasibility rating for the attack via cloud access (related to [WP-15-06]).

6-38): The diagram elements in the attack tree (logic gates, transfer elements, attack steps) are represented by SysML requirements in diagram form. These requirements are represented in the form of a tree structure. In the Attack Feasibility Rating, I use these requirements and the tree structure and use them within a hierarchised requirements table. I take advantage of the fact that requirements can be extended by attributes. I supplement the requirements with the factors of the Attack-Potential-Based Approach and with the possibility to define an AFL. Once an element in the tree has been assessed for attack potential, it can be reused as a referenced object in other attack trees.

Within the hierarchised requirements table (or tree structure), I determine the AFL as follows: (1) The AFL of a requirement is determined based on the AFL of the underlying requirements. (2) In the case of a transfer element, the AFL of the underlying subtree is taken. (3) In the case of an OR gate, the lowest AFL of the underlying requirements shall be used. If an attacker has several ways to achieve the goal, it is sufficient to choose the simplest way. (4) In the case of an AND gate, the highest AFL of the underlying requirements is used. In this case, an attacker must successfully complete several related attack steps. If the attacker cannot perform the attack step with the highest AFL, it is not sufficient to have performed the other attack steps successfully.

In Figure 6-38 I show the attack feasibility rating for the Cloud Access subtree from Figure 6-37. I assume that a DSS is accessible via a content management system over the Internet. I assume that a strong password is not used.

In the following, I describe the steps to systematically guess the credentials using a brute force attack. The attack consists of several simple steps and one advanced step (cf. Elements 8-10 and 5 in Figure 6-38).

In the following, I will explain the evaluation of element 5. This is the preparation for the use of a specialised tool that automatically enters the credentials on a specific website. Expert knowledge is required to find the login fields of the content management software. I assume that the login area has been programmed in a non-trivial way. I assume that the cloud service can be found using a computer search engine. Such search engines store the metadata of servers. This can be used to find servers that use special content management software for DSS. I estimate that this work can be done within a week. Once the address of the server is found, the window of opportunity for the attack is unlimited. Using this information and a mapping table from ISO/SAE 21434 [**ISO21**] for the attack-potentail-based approach, I rate the AFL as medium. Considering all the AFLs of the subtree together, I obtain an AFL rating of medium for cloud access.

#	Name	Attack tree - Feasibility rating	Safety impact - Severity	Financial impact	Operational impact	Privacy impact	Safety risk level	Financial risk level	Operational risk level	Privacy risk level
1	299 Cloud access	Medium	Severe	Severe	Severe	Negligible	4	4	4	1

Figure 6-39: Risk calculation for a damage scenario (related to [WP-15-07]).

6.6.7.3 [WP-15-07] Risk calculation

According to ISO/SAE 21434, the risk value for each threat scenario [WP-15-07] is determined from the impact of the associated damage scenarios and the attack feasibility of the associated attack paths. In general, the following applies Risk = Impact X Feasibility. The risk value of a threat scenario shall be between 1 and 5, with a value of 1 representing minimal risk.

Based on the impact rating from Phase 2 for the Cloud Access threat scenario and the attack feasibility rating from Phase 5, the risk shown in Figure 6-39 for the categories *security, financial, operational and privacy* results. Due to the high impact and medium feasibility, the risk of a cloud access attack is predominantly high.

#	Name	Safety risk level	Financial risk level	Operational risk level	Privacy risk level	Risk treatment option
1	299 Cloud access	4	4	4	1	reduce the risk

Figure 6-40: Determination of a risk treatment (related to [WP-15-08]).

6.6.7.4 [WP-15-08] Risk treatment decisions

For each threat scenario, at least one risk management option [WP-15-08] must be selected, taking into account the risk values determined: (1) Avoidance of the risk by removal of the source. (2) Reduce the risk by applying a countermeasure. (3) Sharing the risk, e.g. by transferring the risk to an insurance company. (4) Reasonable retention of risk. Since the identified risk for the cloud access threat scenario is high, I decided to reduce the risk by using a countermeasure.

#	Name	Asset	Attack tree - Feasibility rating	Cybersecurity assurance level/ CAL
1	196 Ensure the integrity of the traffic sign recognition of speed signs.	Multi purpose camera	High	CAL4

Figure 6-41: Derivation of a cybersecurity goal (related to [WP-09-03]).

6.6.7.5 [WP-09-03/04] Cybersecurity goals/claims

Depending on the risk analysis, cybersecurity goals [WP-09-03] and cybersecurity claims [WP-09-04] need to be defined in accordance with ISO/SAE 21434.

A cybersecurity claim provides a justification for retaining or sharing a risk. These claims are maintained throughout the development process for monitoring purposes. Further evidence from later phases of development may also lead to a change in the basis of the decision.

A cybersecurity goal is a concept level cybersecurity requirement associated with one or more threat scenarios. If the risk treatment decision for a threat scenario involves risk reduction or avoidance, one or more cybersecurity goals must be defined.

To classify different levels of security in the automotive sector, ISO/SAE 21434 introduces the concept of Cybersecurity Assurance Levels (CALs). Their use is optional. A CAL determines the level of rigour with which security activities must be performed. CALs can be applied throughout the product lifecycle and supply chain. The highest level is CAL4 and the lowest is CAL1.

For example, activities related to ensuring that cybersecurity activities have been adequately performed can be assigned to CALs: (CAL1) No cybersecurity assessment is required. (CAL2) Cybersecurity assessments are performed by someone other than the originator. (CAL3) Cybersecurity assessments are performed by someone on a team other than the originator. (CAL4) Cybersecurity assessments are performed by a person who is independent of the originating department in terms of management, resources and approval authority.

In the approach, I use CALs because they facilitate appropriate communication about the rigour of security measures along the product lifecycle.

In the application example, no cybersecurity claim was deemed necessary. Due to the high risk of a cloud attack on the DSS, there is a risk that the multi-purpose camera could be tricked by displaying a traffic sign on a DSS. Therefore, I had to derive the cybersecurity goal shown in Figure 6-41.

The specification of functions at the vehicle level (e.g. traffic sign recognition) is done by an OEM. For this purpose, the application example uses a multi-purpose camera, which is usually developed by a supplier and provided to the OEM. Taking Figure 6-41, I rate the cybersecurity goal as CAL4. CAL4 is met by the supplier if the supplier also performs a cybersecurity assessment. This is because the supplier is independent of the OEM in terms of management, resources and approval authority as described in CAL4.

6.6.7.6 [WP-09-05] Verification report for cybersecurity goals

According to ISO/SAE 21434, verification shall be performed for cybersecurity goals [WP-09-05].

In terms of the approach, this means the following: (1) The results of the Impact Analysis and Feasibility Analysis (Phases 2, 3, 5) must be checked against the Item Definition (Phases 1, 4) for correctness and completeness. (2) The results of the Impact and Feasibility Analysis shall be checked against the Risk Treatment Decisions (Phase 5) for completeness, correctness, and consistency. (3) The Cybersecurity Goals and Cybersecurity Claims (Phase 5) shall be checked against the Risk Treatment Decisions for completeness, correctness, and consistency. (4) The Cybersecurity Goals and Cybersecurity Claims need to be checked for consistency with the Item Definition.

To do this, I use a digital checklist in the approach. For this, I use a SysML requirements table (cf. Figure 6-42). This references all the work products of the concept phase and

assigns the verification criteria to be considered to these work products as attributes. I assign the values Satisfied, Partially Satisfied, Not Satisfied, and Not Required to the verification criteria. If a check criterion is Partially Satisfied or Not Satisfied, a justification is described in an additional attribute.

The digital checklist can be used as a starting point for an auditor. The auditor can (randomly) check whether the referenced work products meet the verification criteria. The end-to-end linking of work products and their deliverables allows design decisions and risk analysis results to be traced through all work products from the concept phase onwards.

#	Work product	Consistency	Correctness	Completness	Comment
1	[WP-09-01] Item definition - Environment level	Not demanded	Satisfied	Satisfied	
2	[WP-09-01] Item definition - Vehicle level	Not demanded	Satisfied	Satisfied	
3	[WP-15-01] Damage scenarios	Not demanded	Satisfied	Partially satisfied	Further damage scenarios possible
4	[WP-15-02] Assets with cybersecurity properties	Not demanded	Satisfied	Satisfied	
5	[WP-15-03] Threat scenarios	Not demanded	Satisfied	Partially satisfied	Further threat scenarios possible
6	[WP-15-04] Impact ratings with associated impact categories	Not demanded	Satisfied	Satisfied	
7	[WP-15-05] Attack paths	Not demanded	Satisfied	Satisfied	
8	[WP-15-06] Attack feasibility ratings	Not demanded	Satisfied	Satisfied	
9	[WP-15-07] Risk values	Not demanded	Satisfied	Satisfied	
10	[WP-15-08] Risk treatment decisions	Satisfied	Satisfied	Satisfied	
11	[WP-09-03] Cybersecurity goals	Satisfied	Satisfied	Satisfied	
12	[WP-09-04] Cybersecurity claims	Satisfied	Satisfied	Satisfied	

Figure 6-42: Verification report for cybersecurity goals [WP-09-05].

6.6.7.7 [WP-09-06] Cybersecurity concept

#	Text	Allocation	Mitigation	Security design pattern
1	When detecting a speed sign using a Multi purpose camera, the context of the speed sign has to be associated using the Advanced Driver Assistance System.	: Multi purpose camera : Advanced Driver Assistance System	238 (M22) Secure external interfaces	281.1 (01) Context box pattern
2	The Advanced Driver Assistance System has to support sensor fusion. For this purpose, the image recognition of speed signs has to be supplemented by a Multi purpose camera and depth information by a Radar sensor.	: Advanced Driver Assistance System Radar sensor : Multi purpose camera	223 (M10) Message verification	280.10 (10) Sensor fusion
3	The Advanced Driver Assistance System must log which traffic sign was detected and from which data sources it was detected (multi purpose camera or radar sensor or both sensors).	: Advanced Driver Assistance System Radar sensor	239 (M23) Cybersecurity best practices	278.3.10 (6.11) Security logger and auditor

Figure 6-43: Assignment of cybersecurity controls to vehicle components and derivation of cybersecurity requirements (related to [WP-09-06]).

According to ISO/SAE 21434, cybersecurity requirements [WP-09-06] for the item and its operational environment shall be described for the defined cybersecurity goals. The

standard requires the description of cybersecurity controls that serve to meet the cybersecurity goals. A cybersecurity control is a security measure that helps prevent cyber-attacks or minimizes the risk of an active attack. UN R155 lists 24 cybersecurity controls to be considered in the context of the approval of a vehicle manufacturer's CSMS. [7]. The cybersecurity concept is formed from the cybersecurity requirements of the item and its operational environment, with associated information about the cybersecurity controls.

In the approach, I use a SysML Requirements Table with specific attributes to describe the cybersecurity requirements (cf. Figure 6-43): (1) Based on the cybersecurity goals, I identify the relevant cybersecurity controls of UN R155. This is done by referring to a requirements table that contains all the cybersecurity controls of UN R155. (2) Since the cybersecurity controls of UN R155 contain only a brief description, I also use security design pattern catalogues, which contain detailed information on the design of the cybersecurity controls. In Sections 5.6 and 5.7, I present security design patterns that can be used in the concept phase. (3) After analyzing the Security Design Patterns, I identify the relevant vehicle components in which the cybersecurity control measures need to be implemented. In Section 5.6, I describe how such a security design pattern can be implemented in a system architecture as part of the concept phase. (4) Based on this, I derive the textual description of the cybersecurity requirements. I proceed in the same way as in Section 5.8. I use the elements identified so far to formulate the requirements. Here it is the identified components that are to receive a countermeasure.

I have identified several cybersecurity controls and security design patterns for the considered cybersecurity goal in refCybersecurity goals.

The context box design pattern [**MCL+14**] is an object recognition design pattern. In the application example, object recognition is primarily implemented by the multi-purpose camera component of the vehicle. The design pattern consists of placing a box around a detected object, e.g. a traffic sign. In addition, the objects outside the box are analyzed and related to the object inside the box. In this way, a traffic sign on a road can be distinguished from a traffic sign on a DSS.

In the context of traffic sign recognition, the sensor fusion design pattern [**Bos23-ol**] allows data from different sensors to be fused. By combining this data, a more accurate representation of the vehicle's environment can be created. For example, the multi-purpose camera is used to capture images of the vehicle's surroundings, while an additional LIDAR sensor is used to collect distance data. I assign this cybersecurity control to the Advanced Driver Assistance System (ADAS). An ADAS collects and processes data from various sensors. The additional use of a LIDAR sensor provides depth information. This makes it possible to distinguish between a normal traffic sign and a traffic sign displayed on a DSS.

The Security Logger and Auditor design pattern [**Fer13**] consists of two parts: The security

[7] The UN R155 refers to these cybersecurity controls as mitigations. As mitigations also include preventive security measures, such as the use of access control techniques, I use the term cybersecurity controls here.

logger is a mechanism that logs security-related events in the system and stores them in a log file. The security auditor regularly checks the log file for unusual activity. It can issue warnings if, for example, a high number of certain events have occurred. Since the ADAS processes data from various sources, it can also be used to log security-relevant events, for example, to evaluate an accident. In relation to the application example, it could be logged on which data sources a traffic sign recognition has been performed. If a traffic sign is detected by the multipurpose camera, but a LIDAR sensor indicates an unusual depth detection because the traffic sign is located on a DSS, the driver should be warned that the traffic sign detection is currently not working reliably.

Based on the cybersecurity controls of UN R155, the security design patterns and the allocation to vehicle components, the cybersecurity requirements shown in Figure 6-43 are derived.

6.6.7.8 [WP-09-07] Verification report of cybersecurity concept

According to ISO/SAE 21434, a verification report for the cybersecurity concept [WP-09-07] has to be created. For this purpose, the cybersecurity requirements have to be checked against the cybersecurity goals with regard to the verification criteria of completeness, correctness, and consistency. In addition, the consistency of the cybersecurity requirements has to be checked against the cybersecurity claims.

Similarly to the verification report for the cybersecurity goals, a digital checklist is used for this purpose (cf. Figure 6-44).

#	Work product	▽ Correctness	Consistency	Completness	Comment
1	[WP-09-03] Cybersecurity goals [WP-09-06] Cybersecurity concept	Satisfied	Satisfied	Partially satisfied	Further damage/threat scenarios possible
2	[WP-09-04] Cybersecurity claims [WP-09-06] Cybersecurity concept	Not demanded	Satisfied	Not demanded	Further damage/threat scenarios possible

Figure 6-44: Verification report of cybersecurity concept [WP-09-07].

6.6.8 SysML profile for ISO/SAE 21434

The models/requirements presented were created using a SysML profile that I created. A SysML profile allows the adaptation of SysML for specific application purposes. In the SysML profile (cf. Figure 6-45), I have integrated the used approaches (Attack Potential Approach, ASIL Risk Classification Scheme, Attack Trees with Boolean Logic, Cybersecurity Assurance Level, UN R155 Threats/Cybersecurity Controls, Security Design Patterns) in the form of specific model constructs, attributes, inheritance relationships, and model relationships [8]. This facilitates the creation of the 15 ISO/SAE 21434 work

[8] Almost all of the elements shown in Figure 6-45 inherit from the SysML Custom Requirement element. I have not shown this relationship in the figure so as not to complicate the figure unnecessarily.

products and increases the reusability of ISO/SAE 21434 models/requirements. This saves time and resources in modeling/specification and increases the efficiency and consistency of the content produced.

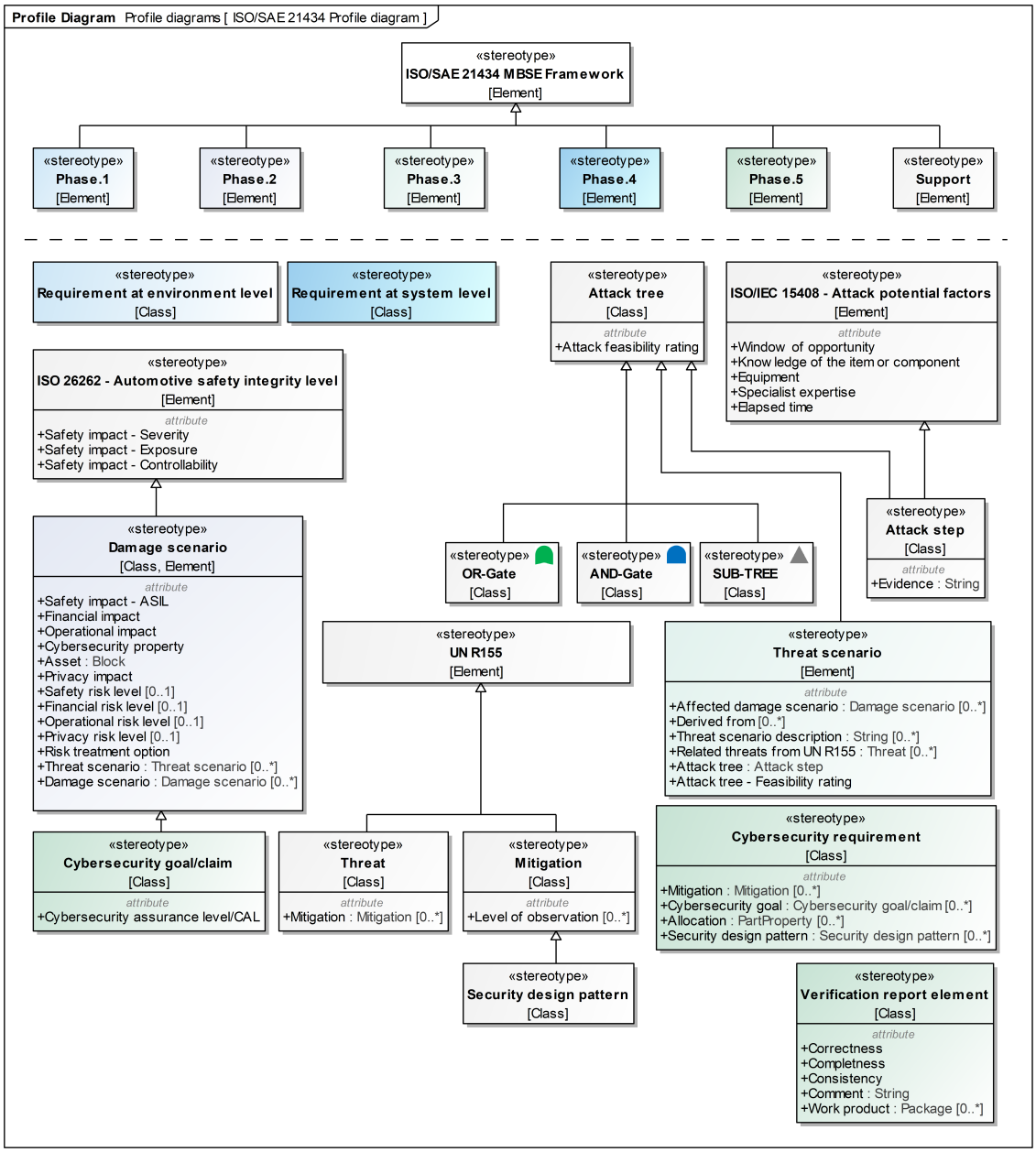


Figure 6-45: ISO/SAE 21434 profile diagram.

6.6.9 Real life test

In this section I describe a real test. In this test, I displayed a traffic sign on a DSS to fool the traffic sign recognition of a vehicle. The credentials for a DSS were provided to me for this purpose. I then describe my investigations to perform a brute force attack in case no credentials are available. For legal reasons I did not perform the attacks.

6.6.9.1 Tricking traffic sign recognition with the help of a manipulated DSS

By conducting the real-life test, I wanted to make sure that the application example was realistic in order to better understand the modelling/specification issues and improve the approach based on that. In the real-life test, I wanted to see if traffic sign recognition could be tricked by displaying a traffic sign on a DSS.

In Figure 6-46 I show the vehicle used and the footage from the real test. The result is as follows: When passing the DSS with the speed sign displayed, the driver is warned acoustically and visually if the speed is higher than the displayed speed. The test vehicle was unable to distinguish the displayed traffic sign from a real traffic sign. In particular, the vehicle did not use the existing LIDAR sensor, e.g. for sensor fusion.

Figure 6-46: Testing the feasibility of the attack by displaying a speed sign on a DSS.

Another test with a traffic sign printed on a DIN A3 page and attached to the posts of a pedestrian bridge using a 2015 Honda CR-V with traffic sign recognition gave the same result. The Honda CR-V had a navigation system with traffic sign information. The car did not use this information in the camera traffic sign detection to get a more accurate result using sensor fusion. As I did not had access to a car with an automatic speed limiter, I could not test the traffic sign recognition together with the speed reduction. For the real scenario, I assume that there is a vehicle following closely behind the front vehicle. I also assume that the car in front has an automatic speed reduction assistant. In this constellation, I assume that a collision would occur in the real scenario.

6.6.9.2 Checking the feasibility of the attack

6.6.9.2.1 Investigations for an attack on DSS of different providers For the access types cloud access, wifi access and physical access, I researched on the internet how DSS can be attacked at each location. In total, I was able to identify and evaluate 28 attack steps (all attack steps are listed in the Appendix in Section B.1). According to the analysis, an attack via physical access is the easiest way. This requires PC knowledge and simple tools to open the DSS. As DSSs are often located on roads, the time window for an attack is very limited. The WLAN attack is the most difficult. It requires a high level of expertise and the presence of a service technician near the DSS to configure it via Wifi.

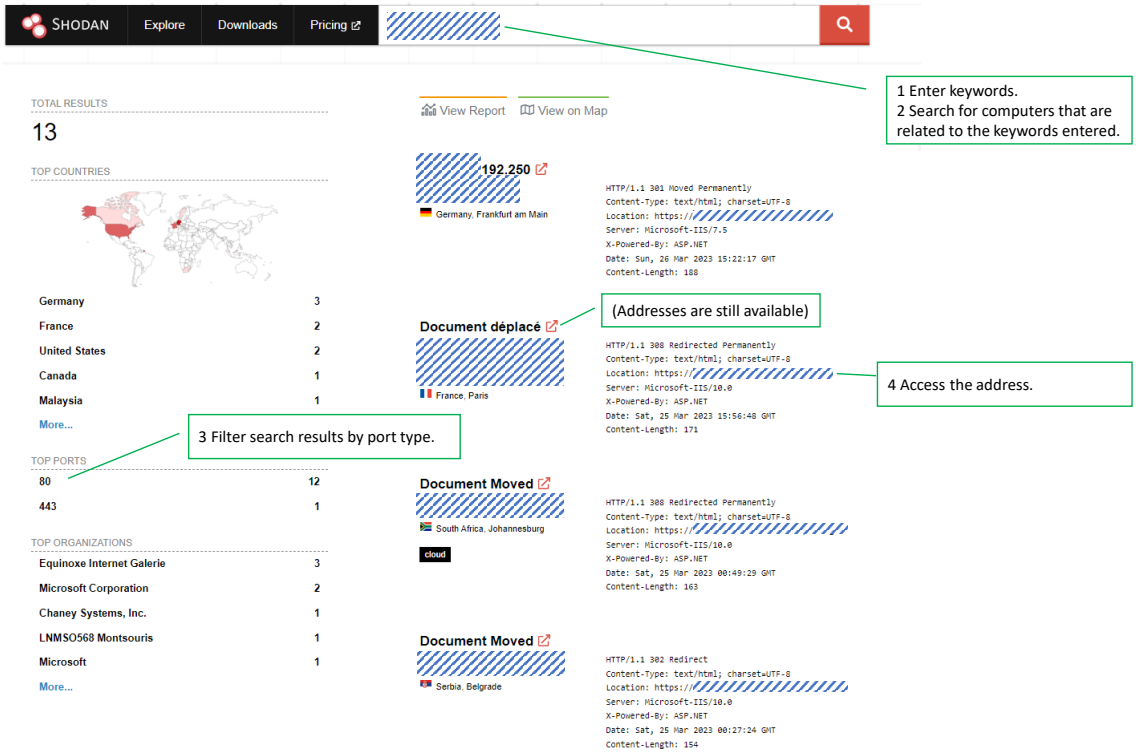


Figure 6-47: Worldwide search for digital signage systems for a specific keyword.

Attacking via cloud access is the best way for an experienced attacker. In the following I will give some details for a possible attack via cloud access. DSS can be found using search engines and specific keywords. In this case, I analysed the user manual of a content management software (CMS) for DSS and noted down keywords. In addition, the user manual provided me with information about the password specification. Using a keyword found in the manual, I was able to find the web addresses of 13 DSS that were accessible via a login page (cf. Figure 6-47).

The password specification gave me information about the use of tools to generate password lists. After analysing the source code of the login page, I was able to locate the login fields (cf. Figure 6-48). This was the basis for using an automated login guessing tool found on the internet. Such an attack is successful when weak passwords are used, e.g. when standard usernames (e.g. User1) are used when creating a user account and weak passwords (e.g. CompanyName_Year) are used without being changed by the user.

Searching for content management software from another providor, I was able to find 620 DSS addresses accessible worldwide. Unfortunately, these were not accessible via a login web page, so the tools I found were not applicable in this case. However, the DSS could be accessed using login credentials via the TCP protocol. In my opinion, this would require more advanced tools for an attack. It would also require more knowledge of network attacks. Therefore, I did not pursued this path any further.

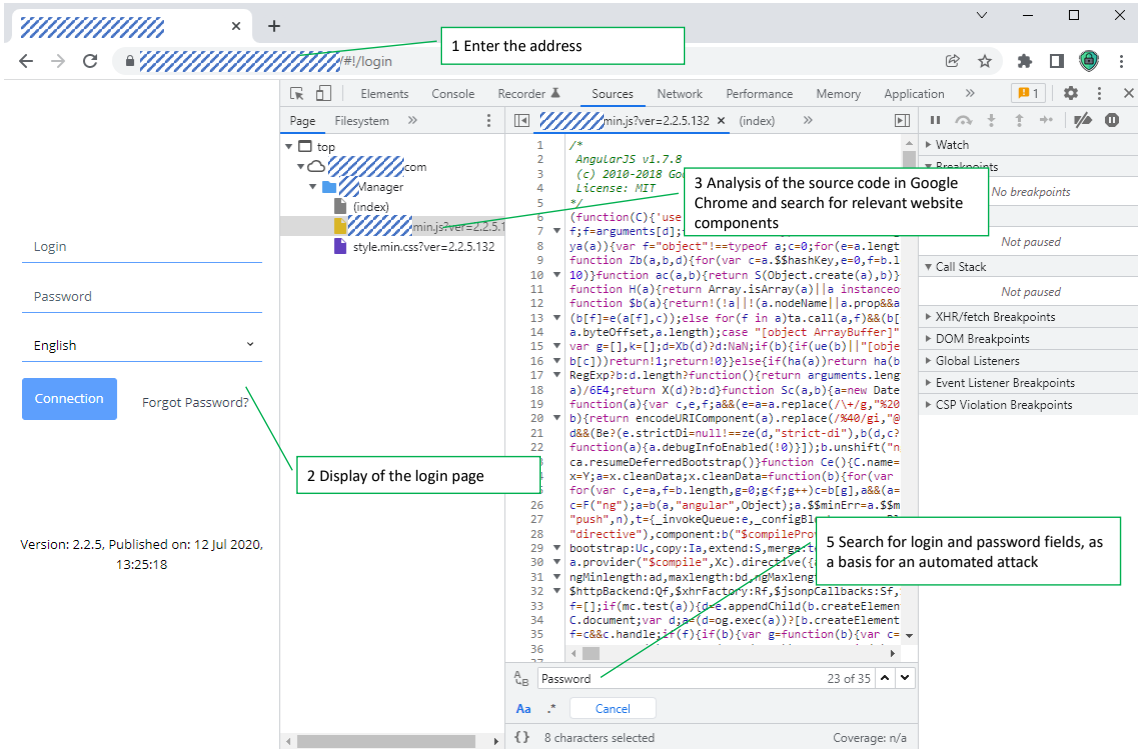


Figure 6-48: Examining a found address and manually searching login and password fields to prepare for an automated brute force attack.

6.6.9.2.2 Investigations for an attack on DSS of a specific provider The infrastructure for the DSS for which I have been given access data is managed by a DSS provider Y in city X. The company website of Y shows photos and exact locations of about 40 DSS (cf. Figure 6-49).

According to the company website, this provider has 1500 customers and 6000 DSSs have been set up. This means that there are at least 1500 accounts with an average of 4 DSSs per customer. Based on the credentials provided to me for one DSS, I could see that there were a total of 5 accounts with access to that DSS. These included the marketing department, the IT department, two test accounts and one account for remote maintenance by the provider. I suspect that the actual number of accounts is many times higher. I suspect that not all IT departments of the 1500 customers use strong passwords when creating additional accounts for the DSS.

The provider's web address, like the DSS addresses found above (in the Shodan computer search enginge), had a generic structure: "Well-known technical term in the field of cloud infrastructure"/"Special prefix" + "Company name" + ".de" + "String to login page". The string to the login page is the same as for the addresses found above. The company name can be found on the Internet. According to my research, I was not able to find the full address using specific keywords ("company name" + "login page string") in a search engine. I suspect that an attacker could find out this address through social engineering (e.g. questions to support/questions in special forums). It might also be possible to guess the address based on the generic structure and by comparing it to addresses found on the

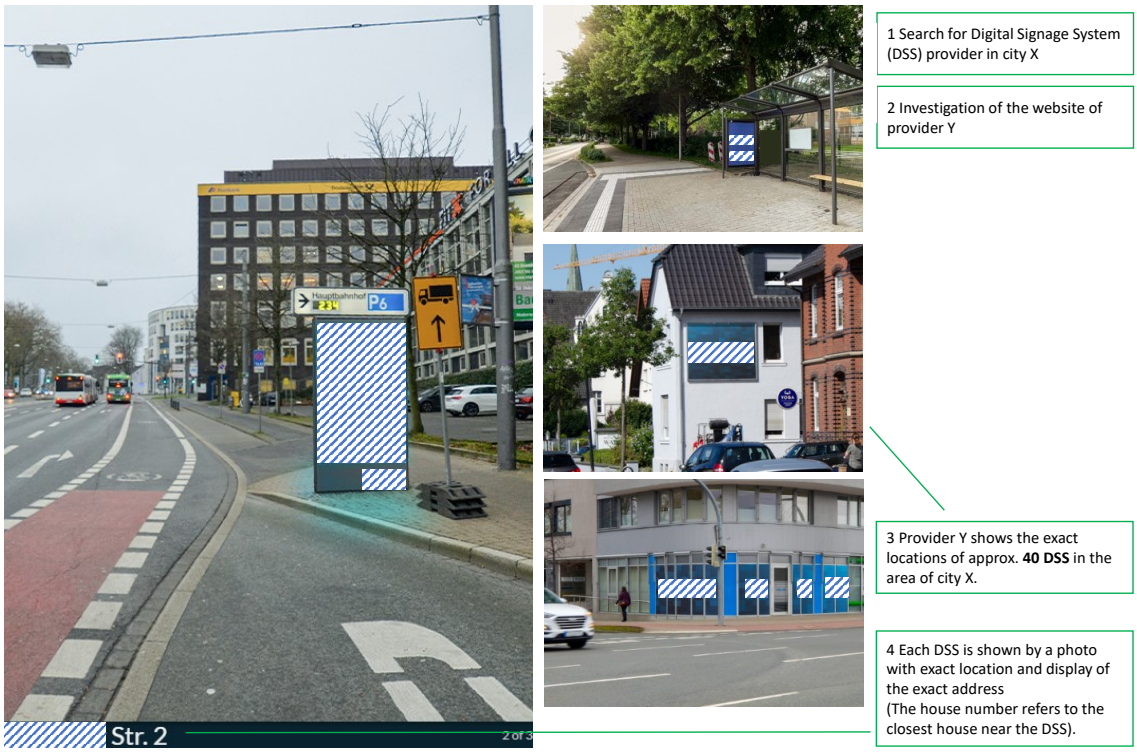


Figure 6-49: Documented locations of digital signage systems (DSS) from a DSS provider.

Internet.

The potential access to so many DSS through a single address could be an attractive target for an attacker. With potentially many vulnerable accounts, potentially many DSS could be manipulated. This would increase the likelihood of collisions resulting from the display of traffic signs on DSS.

6.6.10 Workshops

In three workshops over the course of a year, I presented the approach described in Section 6.6 to automotive security engineering experts from industry (BOSCH, DENSO, PWC) and research. The goal was to get feedback on how to improve the approach.

In addition, I presented the real-life test from Section 6.6.9 and presented the research on the feasibility of the attack. The goal here was to ensure a real-life application example.

In general, the content of the workshops, the real-life test and the attack research were well received. The feedback from Workshop 2 caused me to revise a major part of the solution. I presented the revision in Workshop 3.

6.6.10.1 Workshop 1

The focus of the first workshop was to present the steps from the identification of use cases to the assessment of damage scenarios. These are Phases 1 to 2 of my approach.

Apart from minor details, there was no negative feedback. The active part of the workshop was the evaluation of the damage scenario (cause speed reduction) shown in Figure 6-31. I presented the categories safety, financial, operational and privacy with all levels and detailed textual descriptions. In a survey, the values were entered in relation to the damage scenario. The result was broadly in line with my assessment in Figure 6-31. The damage scenario described was rated as major for the categories safety, financial and operational with regard to the possible collision by closely approaching vehicles. As the vehicles in the described damage scenario were in town and a speed of 70 km/h was assumed, injuries in the case of a collision with fatalities (severity rating severe) were excluded. The privacy impact was chosen to be negligible as the damage scenario had no impact on personal data.

6.6.10.2 Workshop 2 & 3

The second workshop presented the steps leading to the modelling and assessment of possible attacks. In particular, the derivation of cybersecurity objectives and the selection of cybersecurity controls. These are Phases 3 to 5 of my approach. SysML activity diagrams were used to model the attack steps. (I1) From the point of view of experienced risk analysis experts, the use of attack trees was suggested instead, as attack trees are commonly used in practice. At this point, an approach for holistic evaluation of attack paths was used. It was criticised that individual attack steps were not assessed. (I2) The evaluation of individual attack steps allows the reuse of evaluations of these attack steps in different attack trees. At the same time, different attacks, some of which contain the same attack steps, can be better compared because the same evaluation was used for the same attack steps.

In the third workshop, suggestions for improvement I1 and I2 were implemented. There was no negative feedback. This was the basis for the approach described in Section 6.6.

6.7 Evaluation of the work according to the requirements

In this section an evaluation of the developed Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineeringtakes place based on the requirements from Section 2.5. For this purpose, an explanation is given for each requirement as to how these are fulfilled by individual or several components of my work. An overview of the requirements and their relation to the framework I created is shown in Figure 6-50. In Section 7, I describe the limitations of my work and derive future research topics.

R1) Support in the creation of the work products of the concept phase of ISO/SAE 21434: The framework supports the collaboration of several experts from different disciplines, departments and companies in (online) workshops. This is done with the help of a process model (cf. Section 5.9). The process model divides the creation of the 15 work products of the concept phase into several phases (cf. Figure 5-26). Each phase has a differ-

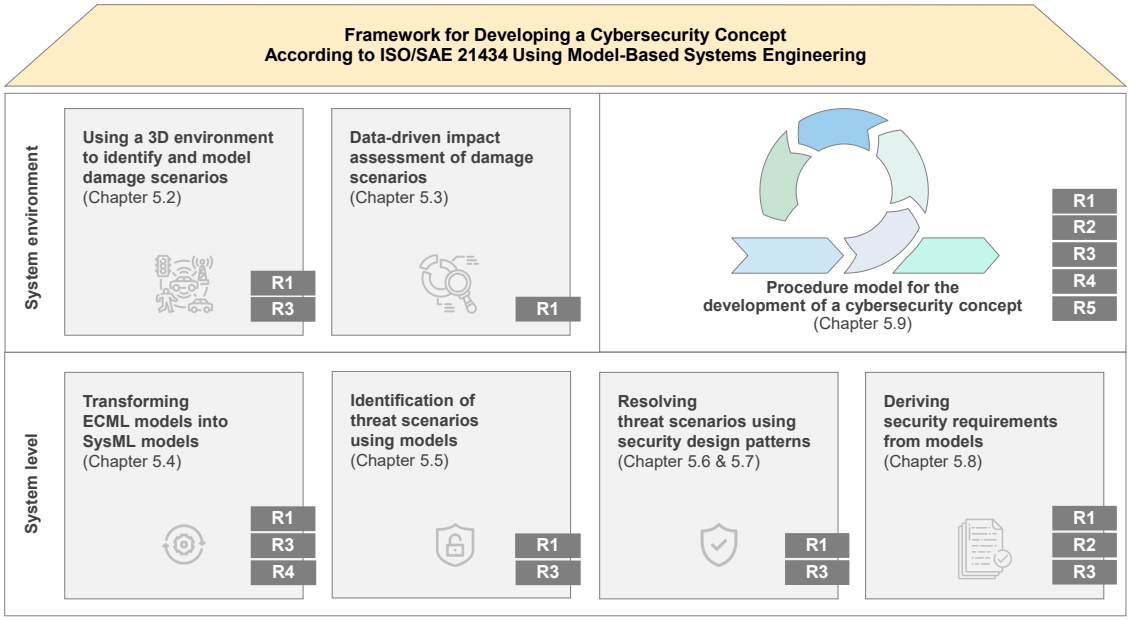


Figure 6-50: Requirements fulfilment using the developed framework.

ent focus. The work products include, in particular, the item definition [WP-09-01] which is the main transition to safety engineering and the TARA [WP-09-02] which belongs to security engineering. ISO/SAE 21434 requires the identification of damage scenarios [WP-15-01]. In the context of my work a 3D environment is used for this (cf. Section 5.2). This facilitates the creation of a common understanding of the damage scenarios between the experts. For identified damage scenarios, an impact rating [WP-15-04] must be performed. Aggregate statistical accident data is used to facilitate the rating (cf. Section 5.3). Based on the damage scnarios, threat scenarios [WP-15-03] have to be identified. For this purpose, an approach has been developed which explains the identification of threat scenarios based on created models (cf. Section 5.5). In order to ensure the consistency of the framework with UN R155, the framework takes into account the threats and mitigations listed in UN R155 (cf. Section 2.1.1). To implement the mitigations based on the models created, an approach to apply security design patterns was developed as part of the early system design (cf. Sections 5.6 and 5.7).

R2) Systematic approach for deriving requirements: To ensure compliance of the framework to the concept phase of ISO/SAE 21434, the framework supports the derivation of cybersecurity goals and cybersecurity requirements (cf. Figure 5-26 [WP-09-03]-[WP-09-07]). This is based on the item definition [WP-09-01] and TARA [WP-09-02] created in the context of the framework. These work products are mainly represented by models in the context of this work. To ensure that the derivation of cybersecurity goals and cybersecurity requirements from models is successful, a corresponding approach was developed (cf. Section 5.8). By deriving cybersecurity goals and cybersecurity requirements, the results of the concept phase can be used outside of workshops. E.g., in detailed system design or for communication between vehicle manufacturers and suppliers.

R3) Use of a standardized modeling language from the fiel of systems engineering: Collaboration during the concept phase is carried out by an interdisciplinary team of mostly leading domain experts, who often have limited modeling knowledge. In the area of Model-Based Systems Engineering, SysML is a de facto modeling language. To facilitate collaboration during the concept phase, the framework uses only selected diagram types of SysML and only a few uncomplicated modeling language constructs. In the context of my industrial projects with a German premium vehicle manufacturer, the modeling language ECML was used in the concept phase. In the detailed system design the company used the modeling language SysML. To reduce the engineering effort and to avoid transfer errors, a tool was developed that automatically transforms ECML models into SysML models (cf. Section 5.4).

R4) Realisation in a professional MBSE tool or based on a professionel MBSE platform: My framework uses modeling software from the MBSE domain to create the 15 interrelated work products of the concept phase. Here, each work product must meet numerous requirements to comply with ISO/SAE 21434. In order to minimize the manual effort to ensure the compliance of the work products to ISO/SAE 21434 and to avoid errors in this context, a special SysML profile and template was created (cf. Section 6.6.8). The SysML profile is used to create ISO/SAE 21434 compliant models and requirements. The template supports the creation of the 15 work products and facilitates review by external reviewers. Further, using the modeling software ensures traceability of changes between identified damage scenarios to cybersecurity requirements across multiple linked work products. The ISO/SAE 21434 compliant SysML profile and template provide the basis for applying the approach in the automotive industry.

R5) Realistic and continuous example from the automotive domain: The framework was evaluated using a realistic and consistent application example from the automotive sector for all 15 work products of the concept phase (cf. Section 6.6). The evaluation took place during several workshops with subject matter experts from the automotive security engineering domain (cf. Section 6.6.10). By evaluating the framework using a continuous application example, it was possible to identify interrelationships in the risk analysis that affect multiple work products. The use of a realistic application example supported the acceptance of the framework and the credibility of the risk assessment by the subject matter experts.

This work thus fulfills all the requirements that were set for the framework. The framework enables the creation of a cybersecurity concept in the context of the concept phase with the help of an interdisciplinary team of subject matter experts using MBSE. The framework was evaluated with domain experts using the continuous application example *Intelligent Speed Assistant (including traffic sign recognition)* for all 15 work products of the concept phase. Finally, the credibility of my work could be increased by the domain experts by a real-life test and by a description of an attack (cf. Section 6.6.9). In the real-life test, the sign recognition of a test vehicle could be tricked by displaying a traffic sign on a manipulated digital signage system (DSS). In order to perform an attack on a DSS, I

described an attack on a DSS using freely available software (without providing critical details).

7 Conclusion and future work

The World Forum for Harmonization of Vehicle Regulations (UNECE WP.29) has issued **UN Regulation No. 155**. It defines uniform conditions for the approval of vehicles with respect to cybersecurity and the cybersecurity management system (CSMS). A CSMS refers to a systematic, risk-based approach in defining organizational processes, responsibilities, and managing risks related to cyber threats to vehicles and in protecting vehicles from cyber attacks. UN R155 will be mandatory in the EU for all new vehicle types from July 2022 and for all newly produced vehicles from July 2024. *This means that vehicles developed without a valid CSMS cannot be registered in the EU.*

ISO/SAE 21434 describes specific requirements for a process framework to ensure cybersecurity in the automotive sector. The implementation of the cybersecurity process framework in the company represents the CSMS required by UN R155. ISO/SAE 21434 describes requirements for activities to create a cybersecurity concept for the concept phase. To create the cybersecurity concept, 15 work products must be created. *The ISO/SAE 21434 describes the requirements for creating the 15 work products, but does not define how these work products have to be created.*

Modern vehicles represent complex, intelligent and connected systems. The development of such systems requires the collaboration of different disciplines. One challenge is the complex collaboration and communication in the concept phase. **Model-Based Systems Engineering** (MBSE) supports the holistic description of complex systems and is used to reduce complexity. In order to describe a system to be developed using models, a graphical modeling language, a modeling method, and modeling software are required. *Only a properly adjusted combination of modeling language, modeling method, and modeling software in conjunction with ISO/SAE 21434, enables effective use of MBSE in the creation of the cybersecurity concept.*

In the context of this work, several relevant approaches were investigated. None of the approaches examined, nor any combination of existing approaches, fully satisfies all of the requirements for a *Framework for Developing a Cybersecurity Concept According to ISO/SAE 21434 Using Model-Based Systems Engineering* identified in this work. One key drawback of several approaches was the lack of a methodical support for deriving cybersecurity goals and requirements from models. Only one approach used a standardized modeling language from the MBSE domain. The other approaches were either not model-based or used a domain-specific language that could only be understood by security or modeling experts, but not by an interdisciplinary team of subject matter experts within the concept phase. Although many approaches used models, only four were implemented in MBSE modeling software. Most approaches were implemented and demonstrated using a realistic application example. Unfortunately, most of the application examples were only partially described with models. Few approaches were able to provide digital consistency between the work products of the concept phase.

The developed framework supports the collaboration of several subject matter experts from different disciplines and departments in (online) workshops using MBSE. An overarching **procedure model** supports the creation of the 15 work products of the concept phase, which together form the cybersecurity concept. The procedure model uses the following support tools:

- In order to form a common understanding of identified damage scenarios, a **3D environment** was developed.
- With the help of **aggregated statistical data**, the impact rating for the damage scenarios has been facilitated.
- In order to reduce the engineering effort and to avoid transfer errors, a **transformation tool** was developed which automatically transforms ECML models into SysML models.
- A method for **identification of threat scenarios** based on models was developed. As required by UN R155, its threat and mitigation catalogs were considered in this method.
- A method for **resolving threat scenarios** using security design patterns (SDPs) was developed. SDPs describe the model-based realization of mitigations.
- A method for **deriving cybersecurity goals and cybersecurity requirements** from models was developed.
- Using a created **SysML profile** and a **template**, ISO/SAE 21434 compliant work products can be created in an MBSE modeling tool. This facilitates the review of the created work products by external reviewers.

The framework was **evaluated** using the continuous application example *Intelligent Speed Assistant (including traffic sign recognition)* . This was done for all 15 work products of the concept phase in three workshops with subject matter experts from the field of automotive security engineering . The credibility of my work by the subject matter experts could be increased by a real-life test and by an investigation of an attack. In the real-life test, the sign recognition system of a test vehicle could be tricked by displaying a traffic sign on a manipulated digital signage system (DSS). To perform an attack on a DSS, I described an attack on a DSS using freely available software.

There is a **need for further research and action** with regard to the use of the framework in the development of modern intelligent and connected vehicles.

In the context of the concept phase, there is a trade-off between being simple to apply and accurately capturing domain knowledge. In the context of my work, I have focused on the applicability of my approach so that as much domain knowledge as possible can be extracted in a team of interdisciplinary mostly leading domain experts without a high time budget for the workshops. This is supported by using fewer diagram types and simple

model constructs. In the future, the precision of the captured domain knowledge could be improved by a **role concept** and an **adapted process model**. This could be used to specify which steps should be performed by the workshop moderator and which elements of a modeling language the moderator should use in the workshop. After the workshop, a modeling expert could enhance the precision of the models with the help of detailed elements of a modeling language in collaboration with the workshop moderator.

In my approach, I use the modeling software Cameo Systems Modeler (CMS), which is well established in the MBSE field. CMS enables comprehensive systems modeling, facilitates communication in large companies and large projects, and enables comprehensive systems analysis. However, such software comes at a cost and is potentially expensive for large organizations with many users. If such software is not available in a company, it must be introduced in a company on at least a partial basis. According to my project experience, this can delay the deployment of such modeling software by several years through price negotiations, establishment of an IT infrastructure and training of employees. For faster application of my approach, a **free modeling tool** could be used. For example, Eclipse Papyrus, which fully complies with the SysML specification and supports the creation of SysML profiles, could be used for this purpose.

On the one hand, the SysML profile I created makes it easier to ensure compliance with ISO/SAE 21434, but on the other hand, such extensions to SysML are not standardized. For a successful application of my approach between different parts of a company or an organization, a defined agreement between the involved business units and its acceptance is required. A **training concept**, could be the starting point for a better cooperation between the involved units.

My work integrates the threats and mitigations to be considered according to UN R155. The understanding of these threats and mitigations in the concept phase requires detailed knowledge in security engineering and computer science. During the concept phase, certain details of the system to be designed are still unknown. Here it would have to be checked which threats and mitigations are relevant for the concept phase. For better use of the remaining threats and mitigations in the concept phase, they should be described in a generally understandable way for an interdisciplinary team of experts.

My work focuses on the concept phase of ISO/SAE 21434 and the associated threat analysis and risk assessment (TARA). The resulting cybersecurity concept provides the basis for conducting more in-depth cybersecurity activities in the context of ISO/SAE 21434 for **detailed system development**. The detailed system development considers the system to be developed more from a domain-specific perspective such as software and hardware development. On this basis, a more detailed TARA can be performed and domain-specific countermeasures can be added and applied.

In the context of my projects, I had no access to detailed information about specific vehicle components or the exact interaction between several vehicle components. I have described the impact of an attack on the vehicle architecture in my application example based on

publicly available information. In the future, the performance of my approach could be studied in more detail by using more precise information on vehicle components and vehicle architectures. This could be studied in the context of further **research projects** or **research transfer projects** with automotive companies.

References

[ADK+20] Anacker, H., Dumitrescu, R., Kharatyan, A. et al. (2020), Pattern based systems engineering - Application of solution patterns in the design of intelligent technical systems, [16th International design conference, Cavat, Dubrovnik, Croatia], https://doi.org/10.1017/dsd.2020.107.

[AHS+19] Ashraf, I., Hur, S., Shafiq, M. et al. (2019), Catastrophic factors involved in road accidents: Underlying causes and descriptive analysis, In: PLOS ONE vol. 14, no 10, https://doi.org/10.1371/journal.pone.0223473.

[AFB+13] Almeida, R.L., Filho, J., Braga, J. et al. (2013), Man, road and vehicle: risk factors associated with the severity of traffic accidents. In: Rev Saude Publica, vol. 47, no 4, https://doi.org/10.1590/s0034-8910.2013047003657. (original language), https://pubmed.ncbi.nlm.nih.gov/24346663/. (translated in english).

[THZ17] Amorim, T., Martin, Ma, Z. et al. (2017), Systematic pattern approach for safety and security co-engineering in the automotive domain, [International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2017), Trento, Italy], https://doi.org/10.14279/depositonce-6924.

[BAS+19] Bolovinou, A., Atmaca, U. -I., Sheik, O. et al. (2019), TARA+: Controllability-aware Threat Analysis and Risk Assessment for L3 Automated Driving Systems, In: IEEE Intelligent Vehicles Symposium (IV), Paris, France, pp. 8-13, https://doi.org/10.1109/IVS.2019.8813999.

[BHJ17] Brown, R., Harman, J., Johnson, D. (2017), Improved Memory Elicitation in Virtual Reality: New Experimental Results and Insights, 16th IFIP TC 13 International Conference, Mumbai, India, September 25-29, 2017, Springer International Publishing, 128-146, https://doi.org/10.1007/978-3-319-67684-5_9.

[BB12] Brüggemann, H., Brember, P.(2012), Grundlagen Qualitätsmanagement Von den Werkzeugen über Methoden zum TQM, Springer Verlag, https://link.springer.com/book/10.1007/978-3-8348-8301-8.

[CGA20] Casado-Sanz, N., Guirao, B., Attard, M. (2020), Analysis of the Risk Factors Affecting the Severity of Traffic Accidents on Spanish Crosstown Roads: The Driver's Perspective, In: Sustainability, vol. 12, no 6, https://doi.org/10.3390/su12062237.

[Cai20] Cai, Q. (2020), Cause Analysis of Traffic Accidents on Urban Roads Based on an Improved Association Rule Mining Algorithm, In: IEEE Access, vol. 8, 75607-75615, https://doi.org/10.1109/ACCESS.2020.2988288.

[CS14] Çelik, A.K., Senger, Ö. (2014), Risk Factors Affecting Fatal Versus Non-fatal Road Traffic Accidents: The Case of Kars Province, Turkey. In: International Journal for Traffic and Transport Engineering, vol. 4, no 3, https://doi.org/10.7708/ijtte.2014.4(3).07.

[CER+16] Chandrasegaran, S, Elmqvist, N., Ramani, K., Vinayak, C.P. (2016), Co-3Deator: A Team-First Collaborative 3D Design Ideation Tool, In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2017: 6581-6592), Denver, CO, May 6-11, 2017, https://doi.org/10.1145/3025453.3025825.

[CDP+19] Cheng, B.H.C., Doherty, B., Polanco, N., et al. (2019), Security patterns for automotive systems, [ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS-C), Munich, Germany], https://doi.org/10.1109/MODELS-C.2019.00014.

[CDP+20] Cheng, B. et al. (2020), Security Patterns for Connected and Automated Automotive Systems, In: Journal of Automotive Software Engineering, https://doi.org/10.2991/jase.d.200826.001.

[CCS+23] Chlup, S., Christl, K., Schmittner, C. et al. (2023), THREATGET: Towards Automated Attack Tree Analysis for Automotive Cybersecurity, In: Information 14, 14, https://doi.org/10.3390/info14010014.

[DES+21] Dobaj J., Ekert D., Stolfa J. et al. (2021), Cybersecurity threat analysis, risk assessment and design patterns for automotive networked embedded systems: A case study, In: Journal of Universal Computer Science, 27 (8), pp. 830 - 849, https://doi.org/10.3897/jucs.72367.

[DME+21] Dobaj, J., Macher, G., Ekert, D. et al. (2021), Towards a security-driven automotive development lifecycle, In: J Softw Evol Proc.,https://doi.org/10.1002/smr.2407.

[DOR16] Dori, D. (2016), Model-Based Systems Engineering with OPM and SysML, Springer, New York, https://link.springer.com/book/10.1007/978-1-4939-3295-5.

[ESH15] Edler, F, Soden, M., Hankammer, R. (2015), Fehlerbaumanalyse in Theorie und Praxis, Teil 1: Theoretische und praktische Grundlagen, Springer Vieweg, Wiesbaden, https://doi.org/10.1007/978-3-662-48166-0.

[Fer13] Fernandez-Buglioni, E. (2013), Security Patterns in Practice: Designing Secure Architectures Using Software Patterns, Wiley, ISBN-13 978-1119998945.

[FGH+15] Florides, C., Gregoriades, A., Hadjicosti, J., Michail, H, Pampaka, M. A (2015), Driving simulator for discovering requirements in complex systems, SummerSim ’15 Proceedings of the Conference on Summer Computer Simulation, Illinois, Chicago, July 26-29, 2015, pp 1-10, https://dl.acm.org/doi/10.5555/2874916.2874919.

[GDE+19] Gausemeier, J., Dumitrescu, R., Echterfeld, J. et al.(2019), Innovationen für die Märkte von morgen – Strategische Planung von Produkten, Dienstleistungen und Geschäftsmodellen, Carl Hanser Verlag, München, 2019,https://www.hanser-elibrary.com/doi/book/10.3139/9783446429727.

[GRS14] Gausemeier, J., Ramming, F.J., Schäfer, W. (2014), Design

Methodology for Intelligent Technical Systems: Develop Intelligent Technical Systems of the Future, Springer, Berlin/Heidelberg, https://link.springer.com/book/10.1007/978-3-642-45435-6.

[GJL+10] Gumienny, R., Jobst, B., Lindber, T., Meinel, C. (2010), Is There a Need for a Design Thinking Process?, In Proceedings of Design Thinking Research Symposium 8 (Design 2010), Sydney, Australia, https://hpi.de/fileadmin/user_upload/fachgebiete/meinel/papers/ Design_Thinking/2010 _Lindberg_Design.pdf.

[Hev07] Hevner, A.R. (2007), A Three Cycle View of Design Science Research, In: Scandinavian Journal of Information Systems: Vol. 19: Iss. 2, Article 4, https://aisel.aisnet.org/sjis/vol19/iss2/4.

[ML06] Howard, M., Lipner, S. (2006), The security development lifecycle, Microsoft Press.

[HKM+21] Husung, S., Kleiner, S., Mahboob, A., Weber, C. (2021), Using model-based systems engineering for need-based and consistent support of the design process, In: Proceedings of the Design Society, 1, https://doi.org/10.1017/pds.2021.598.

[HWL17] Hutchison, N., Wade, J., Luna, S.(2017), The roles of systems engineers revisited, INCOSE Volume. 27, Issue 1, Wiley.

[ISO08] International standards organization (2008), ISO/IEC 21827:2008 Information technology — Security techniques — Systems security engineering — Capability maturity model® (SSE-CMM®), https://www.iso.org/standard/44716.html.

[ISO15] International standards organization (2015), ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes, https://www.iso.org/standard/63711.html.

[ISO21] International standards organization, ISO/SAE 21434 Road vehicles — Cybersecurity engineering (2021), https://www.iso.org/standard/70918.html.

[ISO18] International standards organization (2018), ISO 26262: Road vehicles – Functional safety, https://www.iso.org/standard/68383.html.

[JSB+20] Jalilian, M.M., Safarpour, H., Bazyar, J. et al. (2020), Epidemiology of road traffic crashes in Ilam Province, Iran, 2009–2013, In: BMC Research Notes 13, https://doi.org/10.1186/s13104-020-05366-x.

[KV08] Kuechler, B., Vaishnavi, V. (2008), On theory development in design science research: anatomy of a research project, In: Eur J Inf Syst 17, 489–504, https://doi.org/10.1057/ejis.2008.40

[KBD+17] Khastgir, S., Birrell, S., Dhadyalla, G. et al. (2017). Towards increased reliability by objectification of Hazard Analysis and Risk Assessment (HARA) of automated automotive systems, In: Safety Science vol. 99, part B, 166-177, https://doi.org/10.1016/j.ssci.2017.03.024.

[KDH13] Kaiser, L., Dumitrescu, R., Holtmann, J. et al. (2013),Automatic verification of modeling rules in systems engineering for mechatronic systems,[Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 2B: 33rd Computers and Information in Engineering Conference, Portland, Oregon, USA], https://doi.org/10.1115/DETC2013-12330.

[KSM+20] Kumar, S., Srivastava, M., Kharya, P. et al. (2020), Analysis of risk factors contributing to road traffic accidents in a tertiary care hospital, A hospital based cross-sectional study, In: Chinese Journal of Traumatology, vol. 23, no 3, 159-162, https://doi.org/10.1016/j.cjtee.2020.04.005.

[KC21] Kalair, K., Connaughton, C. (2021), Dynamic and Interpretable Hazard-Based Models of Traffic Incident Durations, In: Frontiers in Future Transportation, vol. 2, https://doi.org/10.3389/ffutr.2021.669015.

[KDA+20] Kim, S.m., Do, Gh., Ahn, J. et al. (2020), Quantitative ASIL Estimation Using Fuzzy Set Theory. In: Int.J Automot. Technol. 21, 1177–1184, https://doi.org/10.1007/s12239-020-0111-y.

[KLB+21] Kern, M. Lui, Betancourt, B.(2021), Model-based Attack Tree Generation for Cybersecurity Risk-Assessments in Automotive, In: IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, pp. 1-7, https://doi.org/10.1109/ISSE51541.2021.9582462.

[KMA21] Kruck B., Munk P., Angermeier D. (2021), Safe and Secure: Mutually Supporting Safety and Security Analyses with Model-Based Suggestions, In: Proceedings - IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW, pp. 172 - 181, https://doi.org/10.1109/ISSREW53611.2021.00061.

[LAO21] Lautenbach, A., Almgren, M., Olovsson, T. (2021), Proposing HEAVENS 2.0 – an automotive risk assessment model, In: CSCS '21: Computer Science in Cars Symposium, http://dx.doi.org/10.1145/3488904.3493378.

[LPP+20] Leonavičienė, T., Pukalskas, S., Pumputis, V. et al. (2020), Investigation of Factors That Have Affected the Outcomes of Road Traffic Accidents on Lithuanian Roads, In: The Baltic journal of road and bridge engineering, vol. 15, no 5, 1-20, https://doi.org/10.7250/bjrbe.2020-15.504.

[LCZ+18] Liu, G.,Chen, S., Zeng, Z. et al. (2018), Risk factors for extremely serious road accidents: Results from national Road Accident Statistical Annual Report of China, In: PLOS ONE, vol 13, no 8, https://doi.org/10.1371/journal.pone.0201587.

[MBL19] Maple, C., Bradbury, M., Le, A.T. et al.(2019), Connected and Autonomous Vehicle Reference Architecture for Attack Surface Analysis, In: Appl. Sci. 9, 5101, https://doi.org/10.3390/app9235101.

[MSB15] Macher, G., Sporer, H., Berlach, R. et al. (2015), SAHARA: A Security-Aware Hazard and Risk Analysis Method, [Proceedings of design, automation & test in europe conference & exhibition (DATE), Grenoble, France],

https://doi.org/10.7873/DATE.2015.0622.

[MN20] Munk, P., Nordmann, A. (2020), Model-based safety assessment with SysML and component fault trees: application and lessons learned, In: Softw Syst Model 19, 889–910, https://doi.org/10.1007/s10270-020-00782-w.

[MMS+20] Martin, H., Ma, Z., Schmittner, C. et al. (2020), Combined automotive safety and security pattern engineering approach, In: Reliability Engineering & System Safety, Volume 198, https://doi.org/10.1016/j.ress.2019.106773.

[MCL+14] Mottaghi, R., Chen, X., Liu X. et al.(2014), The Role of Context for Object Detection and Semantic Segmentation in the Wild, In: IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 891-898, https://doi.org/10.1109/CVPR.2014.119.

[MEM+22] Messnarz, R., Ekert, D., Macher, G. et al. (2022), Experiences with the automotive SPICE for cybersecurity assessment model and tools, In: J Softw Evol Proc., https://doi.org/10.1002/smr.2519.

[MBZ+18] Monteuuis, J.-P., Boudguiga, A., Zhang, J. (2018), SARA: Security Automotive Risk Analysis Method, In: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security (CPSS '18), Association for Computing Machinery, New York, NY, USA, 3–14, https://doi.org/10.1145/3198458.3198465.

[MWZ19] Maidl, M., Wirtz, R., Zhao, T. et al. (2019), Pattern-based modeling of cyber-physical systems for analyzing security, [24th European Conference on Pattern Languages of Programs (EuroPLoP'19), Irsee, Germany], https://dl.acm.org/doi/10.1145/3361149.3361172.

[MS05] Mead, N.R., Stehney, T. (2005), Security quality requirements engineering (SQUARE) methodology, [Proceedings of the 2005 Workshop on Software Engineering for Secure Systems—Building Trustworthy Applications,St. Louis, Missouri,2005], https://doi.org/10.1145/1082983.1083214.

[SYS15] OMG (2015), System Modeling Language V.1.4, OMG, Object Management Group, Needham, Massachusetts, USA,

[UML17] OMG (2017), Unified Modeling Language V.2.5.1, OMG, Object Management Group, Needham, Massachusetts, USA, https://www.omg.org/spec/UML/2.5.1/PDF.

[OG18] Oralhan, B., Göktolga, Z. G. (2018), Determination of the Risk Factors That Influence Occurrence Time of Traffic Accidents with Survival Analysis, In: Iranian journal of public health, vol. 47, no 8, 1181–1191, https://pubmed.ncbi.nlm.nih.gov/30186791.

[PZG+21] Plappert, C., Zelle, D., Gadacz, H. et al.(2021), Attack Surface Assessment for Cybersecurity Engineering in the Automotive Domain, In: 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Valladolid, Spain, pp. 266-275, https://doi.org/10.1109/PDP52278.2021.00050.

[Pol16] Pohl, K. (2016), Requirements engineering: Fundamentals, principles, and techniques, Springer.

[RDG+02] Raptis, D., Dimitrakos, T. Gran, B.A. et al. (2002), Model-based risk assessment - the CORAS approach, [Proceddings of the norsk informatikkkonferanse, Tapir], https://doi.org/10.1007/978-0-387-35612-9_13.

[RAG18] Rehman, S. U. , Allgaier, C.,Gruhn, V. (2018), Security requirements engineering: A framework for cyber-physical systems, [2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2018], https://doi.org/10.1109/FIT.2018.00062.

[Rup14] Rupp, C., die SOPHISTen (2014), Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil, Carl Hanser Verlag.

[SAE16] SAE International (2016), Cybersecurity guidebook for cyber-physical vehicle systems J3061, https://www.sae.org/standards/content/j3061_202112.

[Sch20] Schmitt, N. (2020), Durchgängiges Vorgehensmodell zur Anforderungserfassung für die Entwicklung mechatronischer Systeme im Automobil, Dissertation, Paderborn University, https://digital.ub.uni-paderborn.de/hs/content/titleinfo/3375643.

[SSK21] Schmittner, C., Schrammel, B., König, S.(2021), Asset Driven ISO/SAE 21434 Compliant Automotive Cybersecurity Analysis with ThreatGet, In: Systems, Software and Services Process Improvement, EuroSPI, Communications in Computer and Information Science, vol 1442, Springer, Cham., https://doi.org/10.1007/978-3-030-85521-5_36.

[Sho14] Shostack, A. (2014),Threat Modeling: Designing for Security,Wiley.

[SS17] Shrestha, P.P., Shrestha, K.J. (2017), Factors associated with crash severities in built-up areas along rural highways of Nevada: A case study of 11 towns, In: Journal of Traffic and Transportation Engineering, vol. 4, no 1, 96-102, https://doi.org/10.1016/j.jtte.2016.08.003.

[Sin17] Singh, S.K. (2017), Road Traffic Accidents in India: Issues and Challenges, In: Transportation Research Procedia, vol. 25, 4708-4719, https://doi.org/10.1016/j.trpro.2017.05.484.

[SV20] Sini, J., Violante, M. (2020), A simulation-based methodology for aiding advanced driver assistance systems hazard analysis and risk assessment, In: Microelectronics Reliability, vol. 109, https://doi.org/10.1016/j.microrel.2020.113661.

[SWS18] Skoglund, M., Warg, F., Sangchoolie, B. (2018), In Search of Synergies in a Multi-concern Development Lifecycle: Safety and Cybersecurity, In: Computer Safety, Reliability, and Security, SAFECOMP, Lecture Notes in Computer Science(), vol 11094. Springer, Cham., https://doi.org/10.1007/978-3-319-99229-7_26

[TKA+19] Tekaat, J., Kharatyan, A., Anacker, H. et al. (2019), Potentials for the integration of design thinking along automotive systems engineering focusing

security and safety,[International Conference on Engineering Design (ICED), Delft, The Netherlands], https://doi.org/10.1017/dsi.2019.295.

[UN21] UNECE (2021), UN Regulation No 155 – Uniform provisions concerning the approval of vehicles with regards to cybersecurity and cybersecurity management system, https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security.

[VM14] Valasek, C., Miller, C. (2014), A Survey of Remote Automotive Attack Surfaces, Black Hat USA 2014, https://ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf.

[MGS17] van der Merwe, A., Gerber, A., Smuts, H. (2017), Mapping a Design Science Research Cycle to the Postgraduate Research Report, In: Communications in Computer and Information Science (CCIS), Vol. 730, Springer, Cham, https://doi.org/10.1007/978-3-319-69670-6_21

[WLS+20] Wang, J., Lu, H., Sun, Z. et al. (2020), Investigating the Impact of Various Risk Factors on Victims of Traffic Accidents, In: Sustainability, vol. 12, no 9, https://doi.org/10.3390/su12093934.

[ZPR+22] Zelle, D., Plappert, C., Rieke, R. et al.(2022), ThreatSurf: A method for automated Threat Surface assessment in automotive cybersecurity engineering, In: Microprocessors and Microsystems, Volume 90, 104461, https://doi.org/10.1016/j.micpro.2022.104461.

Online references

[AST18-ol] American Software Testing Qualifications Board, Inc. (2018), Sample Exam: Certified Automotive Tester Foundation Level, https://astqb.org/assets/documents/CTFL_AuT_2018_engl_SAMPLE_PAPER.pdf,last access: 2023-09-03.

[Aut21-ol] Automotive SPICE (2021), Automotive SPICE for Cybersecurity, https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_for_Cybersecurity.pdf, last access: 2023-09-03.

[Ban22-ol] Bankrate (2022), Car insurance - Car recall facts and statistics 2022, https://www.bankrate.com/insurance/car/car-recall-facts-and-statistics/,last access: 2023-09-03.

[BPS20-ol] Bosch (2020),Products and Services, https://www.bosch-mobility-solutions.com/en/,last access: 2023-09-03.

[Bos23-ol] Bosch (2023), Sensor data fusion, https://www.bosch-mobility-solutions.com/en/solutions/sensors/sensor-data-fusion/, last access: 2023-09-03.

[Car16-ol] CareerRide (2016), UML practice test https://www.careerride.com/online-practice-test/UML,last access: 2023-09-03.

[Cer21-ol] CERTX (2021), UNECE WP.29 / R155 – How Cyber Security will impact the automotive market as of June 2022, https://certx.com/automotive/unece-wp-29-r155-how-cyber-security-will-impact-the-automotiva-market-as-of-june-2022/, last access: 2023-09-03.

[CON22-ol] Conceptboard (2022), Conceptboard, https://conceptboard.com/,last access: 2023-09-03.

[COR21-ol] Federal Government Germany (2021), Corona-Arbeitsschutzverordnung, https://www.bundesregierung.de/bregde/themen/coronavirus/verordnung-zu-homeoffice-1841120, last access: 2023-09-03.

[COL22-ol] Collaboard (2022), Collaboard, https://www.collaboard.app/,last access: 2023-09-03.

[CSM20-ol] CONSENS at Smart Mechatronics (2022), https://smartmechatronics.de/consens,last access: 2023-09-03.

[DDL22-ol] Digital Dreams Lab (2022), COZMO 2.0, https://www.digitaldreamlabs.com/products/cozmo-robot,last access: 2023-09-03.

[DIS20-ol] Dortmund International Summer School (2020), https://go-study-europe.de/,last access: 2023-09-03.

[DRA22-ol] draw.io (2022), draw.io, https://app.diagrams.net/,last access: 2023-09-03.

[ESC21-ol] escar (2021), 20th escar Europe - The World's Leading Automotive Cyber Security Conference, ,last access: 2023-09-03.

[Dor20-ol] FH Dortmund (2020), Dortmund international summer school, ,last access: 2023-09-03.

[FIG22-ol] Figma (2022), Figma, https://www.figma.com/,last access: 2023-09-03.

[FSO21-ol] Federal statistical office of Germany (2021), Road traffic accidents, https://www.destatis.de/EN/Themes/Society-Environment/Traffic-Accidents/_node.html,last access: 2023-09-03.

[FUS22-ol] Autodesk (2022), Fusion 360, https://www.autodesk.de/products/fusion-360/overview,last access: 2023-09-03.

[Gol15-ol] Goldman, D. (2015), Chrysler recalls 1.4 million hackable cars, CNN Business https://money.cnn.com/2015/07/24/technology/chrysler-hack-recall/index.html,last access: 2023-09-03.

[GOG22-ol] Google (2022), Docs. https://www.google.de/intl/de/docs/about/, last access: 2023-09-03.

[Gre15-ol] Greenberg, A., Hackers remotely kill a jeep on the highway - with me in it (2015), Wired Online, https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/,last access: 2023-09-03.

[IRE21-ol] Quizlet (2021), International Requirements Engineering Board (IREB): Fundamentals level exam questions, https://quizlet.com/423249994/ireb-fundamentals-flash-cards/,last access: 2023-09-03.

[Jso23-ol] ECMA-404 The JSON Data Interchange Standard (2023), https://www.json.org/json-en.html,last access: 2023-09-03.

[KHR22-ol] Khronos (2022), WebGL, https://www.khronos.org/webgl/,last access: 2023-09-03.

[LSP22-ol] LEGO (2022), LEGO® SERIOUS PLAY®, https://www.lego.com/de-de/themes/serious-play/about,last access: 2023-09-03.

[LUD22-ol] Luzid (2022), Luzid-Chart, https://www.lucidchart.com/pages/, last access: 2023-09-03.

[MIR22-ol] Miro (2022), miro, https://miro.com/,last access: 2023-09-03.

[MOD22-ol] modelo (2022), modelo, https://modelo.io/,last access: 2023-09-03.

[MSO22-ol] Microsoft (2022), Office 365, https://www.office.com/,last access: 2023-09-03.

[Nom23-ol] No Magic (2023), Glossary of SysML concepts, https://docs.nomagic.com/display/SYSMLP190/Glossary+of+SysML+concepts, last access: 2023-09-03.

[OMG23-ol] Object Management Group (2023), What is SysML?, https://www.omgsysml.org/what-is-sysml.htm, last access: 2023-09-03.

[ONS22-ol] Onshape (2022), Onshape, https://www.onshape.com/en/,last access: 2023-09-03.

[Pew17-ol] Pew Research Center (2017), What the public knows about cybersecurity, https://www.pewresearch.org/internet/2017/03/22/what-the-public-knows-about-cybersecurity/,last access: 2023-09-03.

[Pol22-ol] Polestar (2022), Polestar 2 manual, https://www.polestar.com/us/manual/polestar-2/2022/, last access: 2023-09-03.

[PUN22-ol] Unity Technologies (2022), Photon unity networking, https://www.photonengine.com/pun,last access: 2023-09-03.

[QUE22-ol] Meta (2022), Occulus Quest, https://www.oculus.com/,last access: 2023-09-03.

[SIM22-ol] dSPACE (2022), SIMPHERA, https://www.dspace.com/de/gmb/home/news/simphera.cfm,last access: 2023-09-03.

[SKE22-ol] Sketchfab (2022), Sketchfab, https://sketchfab.com/,last access: 2023-09-03.

[Sta22-ol] Federal Statistical Office of Germany (2022), Accident Atlas https://unfallatlas.statistikportal.de/,last access: 2023-09-03.

[Str19-ol] Microsoft (2019), Uncover Security Design Flaws Using The STRIDE, https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach,last access: 2023-09-03.

[SVL22-ol] LG (2022), SVL Simulator, https://www.svlsimulator.com/,last access: 2023-09-03.

[Tes23-ol] Tesla (2023), Model 3 owner's manual, https://www.tesla.com/ownersmanual/model3/en_us/, last access: 2023-09-03.

[TU22-ol] TÜV Thüringen (2022), ISO/SAE 21434 – Standard zur Cybersecurity im Automobilbereich, 2022, https://tuev-thueringen.de/blog/iso-sae-21434-standard-zur-cybersecurity-im-automobilbereich,last access: 2023-09-03.

[TUEV22b-ol] TÜV Nord (2022), Fahrassistenzsysteme - Pflicht ab dem 6. Juli 2022, https://www.tuev-nord.de/de/privatkunden/ratgeber-und-

tipps/technik/fahrassistenzsysteme, last access: 2023-09-03.

[THI22-ol] Autodesk2 (2022), Thinkercad, https://www.tinkercad.com/,last access: 2023-09-03.

[TRI22-ol] Trimble (2022), SketchUp, https://www.sketchup.com/,last access: 2023-09-03.

[VEC22-ol] Vectary (2022), Vectary, https://www.vectary.com/,last access: 2023-09-03.

[UIA20-ol] Project References on the UNITY Innovation Alliance (2020), https://www.unity-innovation-alliance.com/en/,last access: 2023-09-03.

[WHO23-ol] World Health Organization (WHO) (2023), Coronavirus disease (COVID-19) pandemic, https://www.who.int/emergencies/diseases/novel-coronavirus-2019,last access: 2023-09-03.

[Win18-ol] Nature Manufacture (2018), Windridge City https://naturemanufacture.com/windridge-city/,last access: 2023-09-03.

Research and teaching projects

[SFC20] Security for Connected Automated CARs (SecForCARs) (2018-2020) is a project funded by the German Federal Ministry of Education and Research that addresses issues related to the security of future connected, autonomous vehicles. The project involved 14 partners. Role: I joined the project shortly after it started as a research associate in 2019. At the beginning, I was one of three team members in the project. As the project progressed, I became the project manager. As project manager, I was the main contact for the project partners and the project executing organization.

[SAV22] Securing Automated Vehicles - Japan-Germany (SAVE) (2021-2022) is a project funded by the German Federal Ministry of Education and Research that extends the topics addressed in SecForCARs from a system-of-systems perspective. The project involved 7 partners from Germany and 4 partners from Japan. Role: Project manager.

[CON23] Connectivity und Resilienz für automatisierte Fahrfunktionen in Deutschland (ConnRAD) (2023-2025) is a project funded by the German Federal Ministry of Education and Research. The goal of ConnRAD is to create systemarchitectures and communication protocols and mechanisms to increase reliability and resilience for connected, safety-critical driving functions. The project consists of 9 partners. Role: Lead of a work package in the project consortium, project manager.

[UPB20] Lecture Model-Based Systems Engineering by Prof. Dr. Roman Dumitrescu at the University of Paderborn (summer term 2020). Role: Preparation and management of an 8-week project with more than 130 master students from the fields of computer science, computer engineering and business informatics[1]. Basis for the student project, were approaches developed by me from the project SecForCARs.

[UPB21] Lecture Model-Based Systems Engineering (summer term 2021). Description similar to the lecture of the summer term 2020, but with 140 master students and based on approaches that have been further developed or newly developed in the projects SecForCARs/SAVE[1].

[1] Consideration of the data privacy ethics of the University of Paderborn: The results of the students were evaluated and anonymized. There was no objection to the usage of the results. In particular, the students were informed that an objection to sharing the results, would have no negative consequences for these students.

Supervised student works

[Kum19] Malle, N.K. (2019). Cybersecurity for cyber-physical vehicle systems according to SAE J3061, Seminar paper, University of Paderborn.

[Rog20] Rogage, R. (2020), Survey of vehicle architectures with focus on security, Seminar paper, University of Paderborn.

[JAR20] JARVIS for Model-Based Systems Engineering (2020), Project group, University of Paderborn.

[Sch20] Schmidt, S. (2020), Entwicklung und Integration eines Voxel-Editors in 3D-Engineer, Practice semester report, Hamm-Lippstadt University of Applied Sciences.

[Kor20] Korb, B. (2020), Virtual-Reality-basiertes kollaboratives Modellieren von Anwendungsfällen in einer dreidimensionalen Umgebung am Beispiel der Software 3D-Engineer, Project work, Hamm-Lippstadt University of Applied Sciences.

[Hag20] Hagemeister, M. (2020), Entwicklung eines Rahmenwerkes für die Bestückung des FieldPower®-Gehäusesystems mit elektronischen Komponenten unter Beachtung des thermischen Verhaltens, Master thesis, University of Paderborn.

[Tis20] Tissen, D. (2020), Methoden-Tailoring für Model-based Systems Engineering, Master thesis, University of Paderborn.

[Sch21a] Schmidt, S. (2021), Web-basiertes kollaboratives Modellieren von Anwendungsfällen in einer dreidimensionalen Umgebung am Beispiel der Software 3D-Engineer, Bachelor thesis, Hamm-Lippstadt University of Applied Sciences.

[Sch21b] Schmidt, S. (2021), Netzwerk-basiertes kollaboratives Modellieren von Anwendungsfällen in einer dreidimensionalen Umgebung am Beispiel der Software 3D-Engineer, Project work, Hamm-Lippstadt University of Applied Sciences.

[3DE21] 3D Environment Based Intelligent Systems Engineering of Advanced Systems (2021), Project group, University of Paderborn.

[Fah21] Faheem, F. (2021), Method for resolving security & safety threats using solution patterns taking into account ISO 21434, Master thesis, University of Paderborn.

[Var22] Varkey, K. (2022), Usage of google maps and statistical data in order to identify critical damage scenarios in a 3D environment, Master thesis, University

of Paderborn.

[Ama22] Amaya, A. (2022), Usage of attack databases in automotive model-based systems engineering in accordance to ISO/SAE 21434, Master thesis, University of Paderborn.

Industry projects

[OEM19a] Use of model-based systems engineering for the advanced development of a sensor system (2019)[2], Customer: German farm machinery group, Role: Support in planning and conducting, Duration: 2 months.

[OEM19b] Analysis and evaluation of a corporate standard with regard to model-based systems engineering (2019)[2], Customer: German vehicle manufacturer, Role: Planning and execution, Duration: 3 months.

[OEM20] Model Formalization I: Analysis and feasibility study regarding the automatic generation of SysML models, based on digitized workshop results in compliance with a corporate standard (2020)[2], Customer: German vehicle manufacturer, Role: Planning and leadership of the development work, Duration: 3 months.

[OEM21a] Model Formalization II: Extended prototype for the automatic generation of hierarchized SysML models (2021)[2], Role: Planning and leadership of the development work, Duration: 3 months.

[OEM21b] Development of a prototype for the selection of consulting services in systems engineering (2021)[2], Customer: German vehicle manufacturer, Role: Planning and leadership of the development work, Duration: 3 months.

[OEM22a] Model Formalization III: Operational data driven validation (2022)[2], Role: Planning and leadership of the development work, Duration: 3 months.

[OEM22b] Analysis and consulting on the use of model-based systems engineering (2022)[2], Customer: German manufacturer of automation and electrical connection technology, Role: Support in the development of an internal guideline for the use of SysML, Duration: 3 months.

[OEM22c] Model Formalization IV: Preparation for intensive validation based on comprehensive operational data (2022)[2], Role: Planning and co-leading of the development work, Duration: 2 months.

[ACA22] Planning, preparation and execution of 8 trainings in the context of the Fraunhofer IEM Academy (2019-2022)[2], Customers: A consulting company, an external Fraunhofer institute, a German supplier from the automotive and mechanical engineering industry.

[OEM23] Identification, formalisation and restructuring of the test process for the use of HIL systems at vehicle level (2023)[2], Role: Planning and conducting the project, Duration: 2 months.

---

[2] For the purpose of confidentiality, the descriptions were generalized and the companies anonymized.

A Supplements to the framework

In Section A.1 I present an extension to the 3D environment from Section 5.2 and to the aggregated statistical data from Section 5.3. This extension integrates Google Maps and accident data from the German Federal Statistical Office into the 3D environment. This reduces the effort to create a road system for modeling damage scenarios. By integrating the accident data, critical road sections can be identified. Based on the road network of Google Maps and the accident data, the degree of realism of the Damage Scenarios can be increased.

In Section A.2 I present an initial security design pattern catalog for use in the concept phase. This catalog was used in an 11-week project with 140 master's students (cf. Section 6.4). The students were from Computer Science, Business Informatics and Computer Engineering. In total there were 28 teams of 5 people.

A.1 3DE extension: Data-driven modeling of damage scenarios

As part of a master's thesis [**Var22**] supervised by me, the 3D environment from Section 5.2 was extended to include the integration of Google Maps and accident data from the Federal Statistical Office. The 10 million accidents registered by the police on German roads were displayed with GPS accuracy in the Google Maps extension of 3DE. This enabled data-driven modeling of damage scenarios.

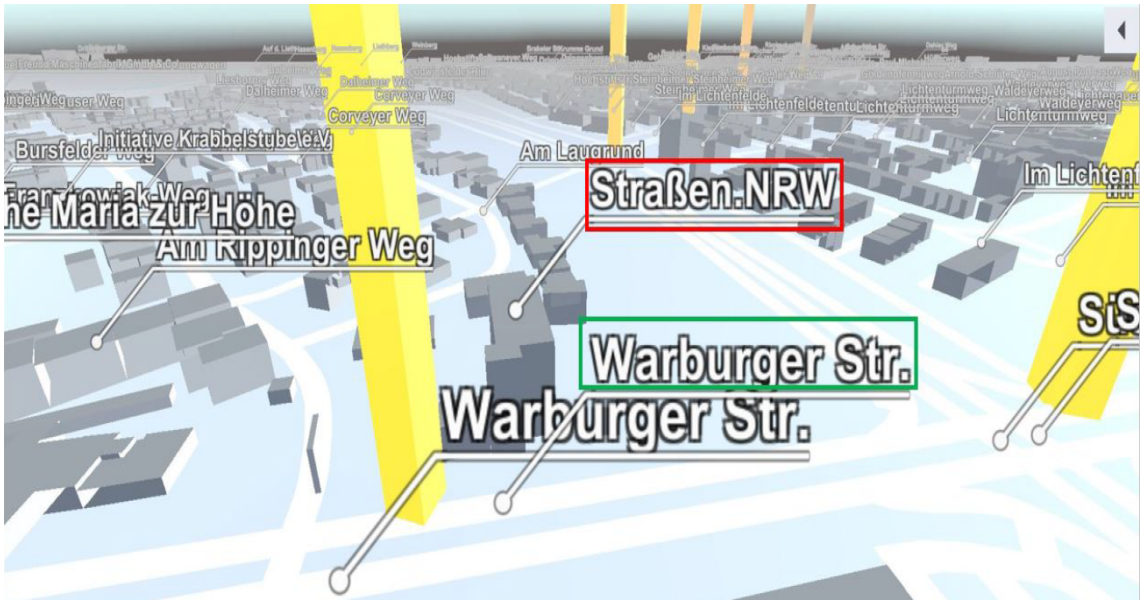


Figure A-1: Integration of Google Maps map data in 3DE.

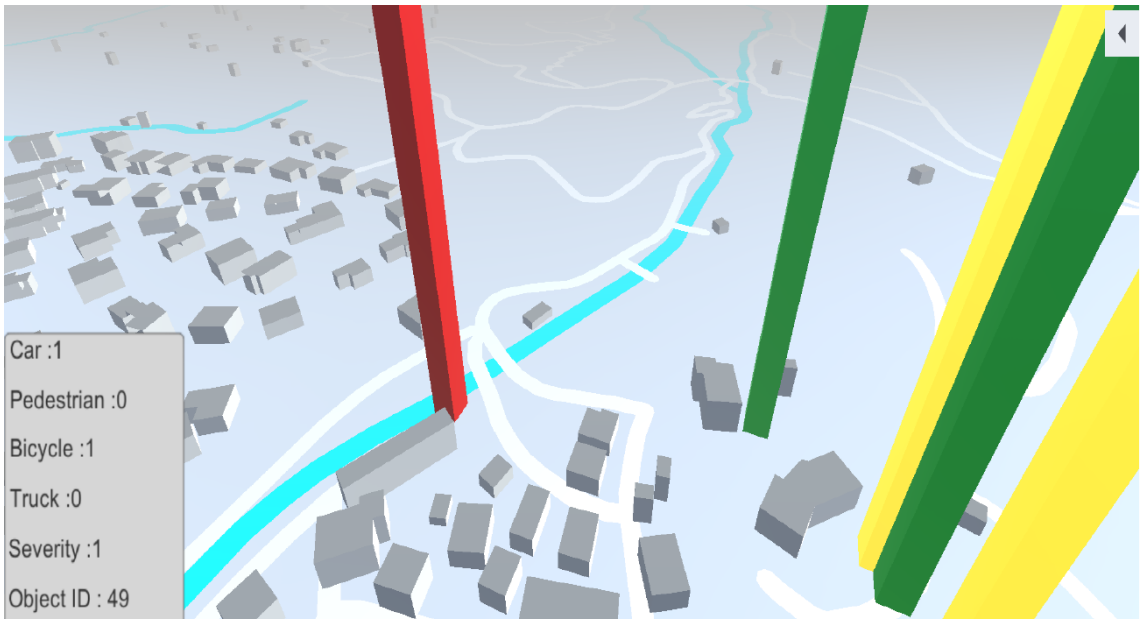


Figure A-2: Integration of accident data of the Federal Statistical Office of Germany in 3DE.

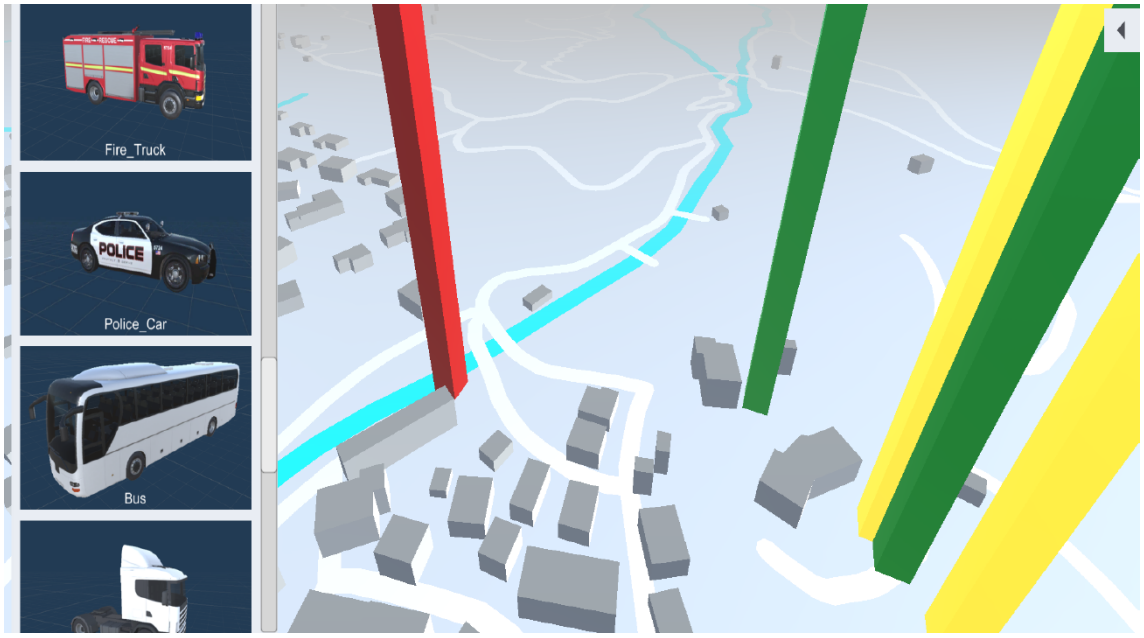


Figure A-3: Selection of 3D objects for placement in 3DE.

Figure A-4: Modeling a damage scenario in 3DE.

Figure A-4 shows a step of a damage scenario in 3DE. The damage scenario contains the following steps: A hacker uses a vulnerability of a digital billboard. This allows the sudden display of arbitrary content on the billboard. The hacker uploads a video containing a traffic sign to the billboard. A vehicle V1 with an intelligent speed assistant detects the traffic sign and abruptly reduces its speed. This offers the potential for a collision by another vehicle V2, which could be behind V1.

A.2 Initial Security Design Pattern Catalogue

In this section, I present 10 SDPs (cf. Figures A-5 and A-6). As a basis, I use the SDPs of [**CDP+20**], which have already been initially agreed with automotive companies. In contrast to Chengs work, I aim to make the SDPs applicable in early system design, thus satisfying R1 and R2 requirements in particular. The SDPs from Cheng have not yet been evaluated. In Section 6.4 I report on an initial evaluation of the SDPs, based on a student project. To describe the SDPs, I use the SysML diagrams Internal Block Diagram (IBD) and Sequence Diagram (SD). I use SysML because it is the de facto standard modeling language in MBSE [**DOR16**]. IBDs are shown in Figures A-5 and A-6 on the left and SDs on the right. IBDs are used to describe structural relationships and SDs are used to describe sequences. SysML provides different diagram types for modeling behavior. I chose SDs because the relationship between IBDs and SDs can be communicated in a simple way since the same blocks are used. For simplicity, I omitted stereotypes in this work and use a color scheme to distinguish elements. System elements are shown in blue and elements that interact with the system are shown in yellow.

[01] Authorization *Problem:* The unauthorized access by an unauthorized subject[1] constitutes a security risk. *Solution:* This can be prevented by authorization. With an authorization, access to resources is managed or controlled and these resources are protected from unauthorized access by subjects. By using a privilege manager, access to the resource to be protected can be managed. The subject's request is either approved or denied after being checked by the privilege manager. *Example:* The privilege manager denies the request if a hacker or any subject tries to access the system's protected object for which it does not have permission.

[02] Blacklist & whitelist *Problem:* In certain constellations, very simple protection mechanisms are needed to access a service. For example, if no powerful hardware is available, or if there are very many users and the manual administration effort must be kept low. *Solution:* In such cases lists can be used. A blacklist prevents access by malicious or untrusted sources. Systems which are whitelisted can be trusted and access is granted. *Example:* Modern vehicles have an infotainment system which allows access to the internet. In a whitelist, the address to a website with software from the vehicle manufacturer can be entered. In a blacklist, known websites which contain harmful programs can be blocked.

[03] Intrusion detection system (IDS) *Problem:* In certain constellations, knowledge about exiting attacks from the past, can assist in detecting attacks. *Solution:* Usage of a database, which contains known attack patterns. A request to a service is compared to the patterns in the database and rejected if the request is marked as a dangerous attack. If a request is not in the database and this request leads to an undesired behavior, the request is stored in the database as an attack. *Example:* A component within a network continuously sends slightly changing login requests to another component in order to gain access to a service. Due to the number of changing login requests, a brute force attack is detected as login data is tried to be guessed.

[04] Tamper resistance *Problem:* Tamper resistance is the problem of detecting, protecting, mitigating or monitoring unauthorized changes on a system or a component. Unauthorized changes can result in vulnerabilities and dangerous system behavior. *Solution:* Tamper Resistance pattern is simply an interface (i.e. termed as a tamper resistance interface) between the subject and tamper-resistant object. This interface has a working state that has by default untempered status of the tamper-resistant object. If someone tries to change the tamper-resistant object then the working state predicts this change that results in breaking the interaction with the tamper-resistant object. *Example:* The vehicle owner wants to install new software via the vehicle's entertainment system (DVD). The installed software may contain malicious intent or vulnerabilities that can alter any vehicle component. A tamper resistance pattern prevents these unauthorized component changes by observing its working state resulting in the breaking interaction with the component due to which the software fails to install within the vehicle system.

---

[1] Subject: person or a technical system

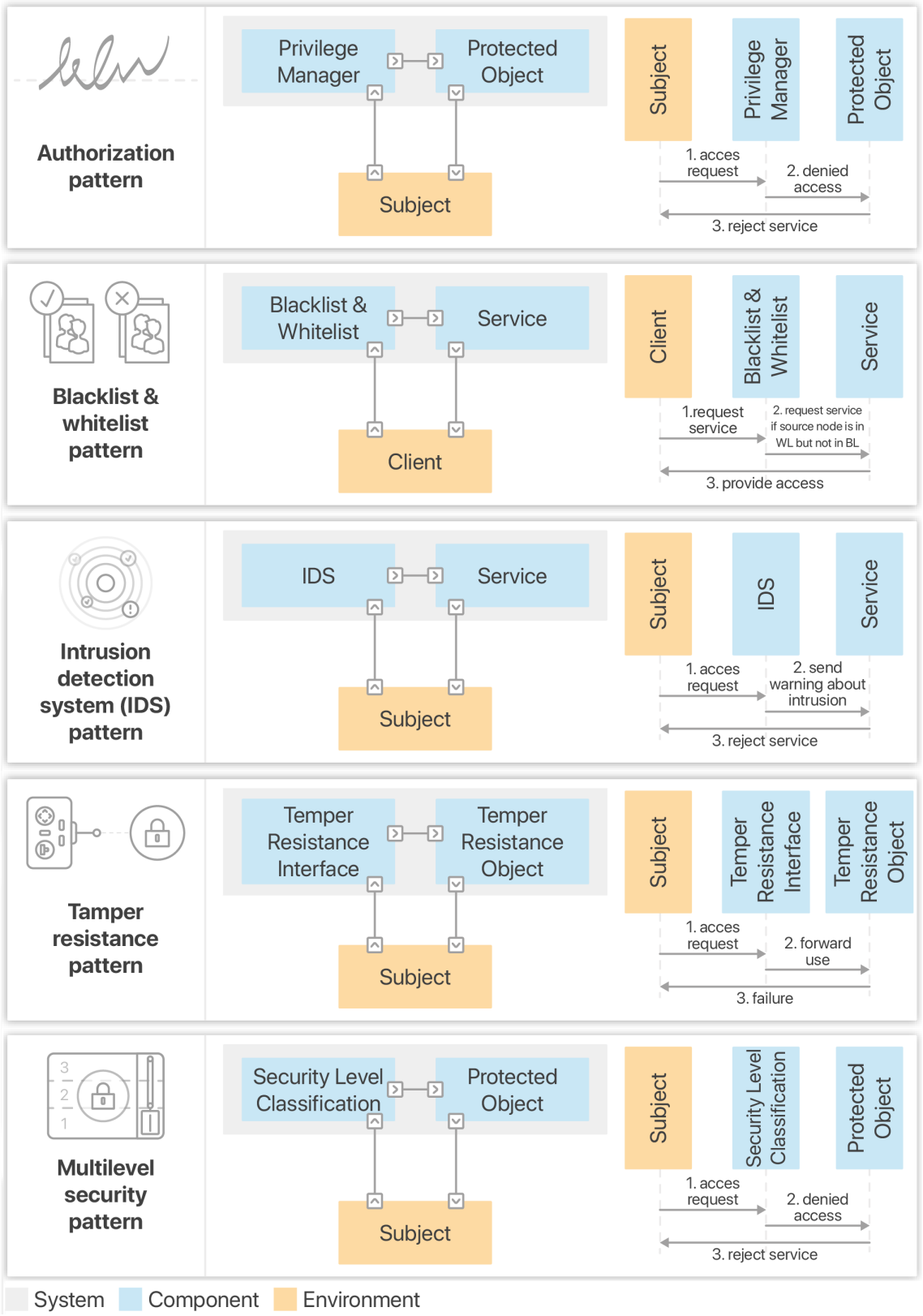


Figure A-5: Initial security design pattern catalogue - Part 1

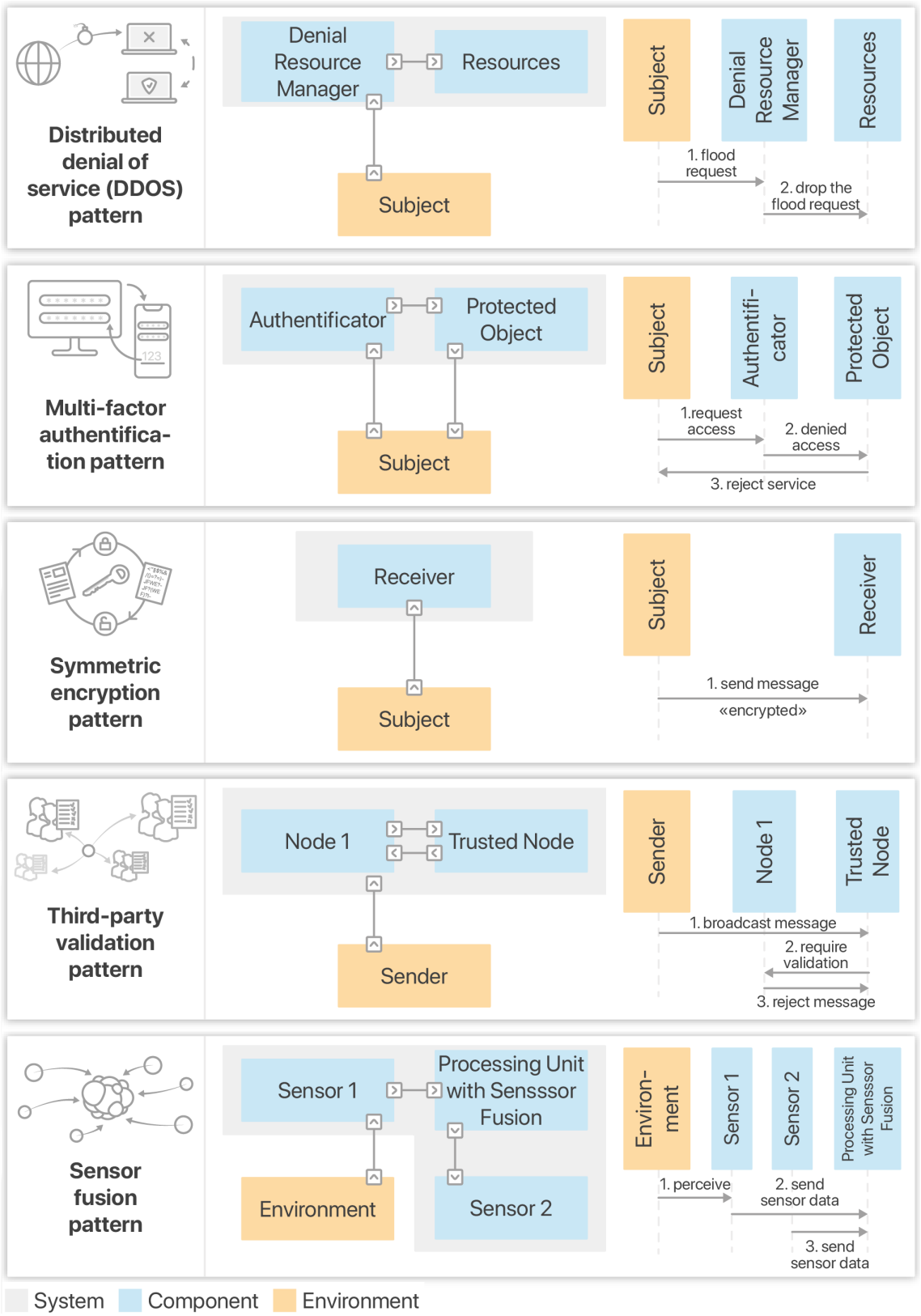


Figure A-6: Initial security design pattern catalogue - Part 2

[05] Multilevel security *Problem:* The problem is about the decision of access in a system with a different security classification. *Solution:* This design pattern proposes access management procedure in a system with the security classification of different levels. The subject will request to access a protected object through a checkpoint. Checkpoint get the security level classification of both object and requested subject. If the subjects security level classification is equal to or greater than the objects security level classification, access will be granted otherwise denied. *Example:* Messages from external communication interfaces such as telematics systems are assigned to lower trust groups to safe internal ECUs or core systems such as ABS brake system.

[06] Distributed denial of service (DDoS) redundancy *Problem:* DDoS attacks flood malicious requests to resources of a network due to which that service is inaccessible to users. These types of attacks work like a jammer that results in some serious consequences. *Solution:* DDoS redundancy is used to protect the network and its resources from DDoS attacks. The mechanism behind this is to provide redundant resources when the particular resource is overburden through service requests. The subject is an entity that sends a message request to the Check Point object. Checkpoint forwards the request to the present resource and also notify the Resource Manager regarding the request. The system's redundant Resources are monitored by the Resource Manager and manage with both the Resources and Check Point to balance the loads. *Example:* Vehicles can still communicate if one of the communication channels of connected vehicles is down by switching to other communication resources.

[07] Multi-factor authentication *Problem:* In case that one of the credentials associated with messages or actors within a system is compromised then another authentication level must exist to prevent the system from attacks. *Solution:* The subject will request access to credentials from the protected object. Between the subject and protected object is an authenticator interface that applies two levels of authentication to get credentials from the protected object. *Example:* A platoon of autonomous vehicles is driving together and communicating with each other via a vehicle-to-vehicle (v2v) network. A new vehicle wants to join a platoon and only gets admitted after successfully passing two layers of authentication by the network.

[08] Symmetric encryption *Problem:* Important information must not be read by all systems or components because, for example, this information is safety-critical. *Solution:* By using encryption between sender and receiver, only these two can read the information. First of all, the sender applies encryption on information using the key that will give cypher information. The receiver will apply decryption on information to get correct information using the same key. *Example:* A sender is sending a message by encrypting the information using symmetric encryption. The attacker acting as a man in the middle is unable to manipulate the sender's information.

[09] Third-party validation *Problem:* A compromised node in the network may send false or malicious messages to other network nodes, resulting in unwanted behavior. *Solution:*

A sender sends a message to a network of nodes. A receiver trusts the message if the following conditions hold: There is at least one other node in the network that considers the sender to be trustworthy. There is a trust relationship between the receiver and the other node. *Example:* Two vehicles V1 and V2 drive behind each other on the highway. The rear vehicle V2 trusts the messages from V1. Now V1 and V2 receive a message from a preceding accident vehicle V0. V1 recognizes V0 on the road side in advance and thus trusts the message from V0. Because of V2's trust in V1, V2 also trusts V0's message.

[10] Sensor fusion *Problem:* The correct perception of the environment is necessary for safe and secure autonomous vehicle driving. Every sensor has its strengths and weakness. A single sensor is limited in terms of providing accuracy, fault free sensor data and enough information for autonomous vehicles for driving purposes. Therefore, fusions of different sensors in the form of sensor fusion are required. *Solution:* Sensor fusion is the fusion of multiple sensors that work on the concept of compensation of the weakness of one sensor with the strength of another sensor resulting in a safe and secure system. Sensor fusion provides accurate results by combining the layers of data from different sensors resulting in better results. *Example:* A camera can be used to detect obstacles on the road. This enables object detection, e.g. pedestrians on the road can be detected. Unfortunately, unlike radar sensors, cameras do not provide precise depth detection. If only one camera is used for object detection, person imprints in the road environment can be identified as pedestrians. The additional use of a radar sensor increases the quality of object detection by providing additional depth information.

B Supplements to the evaluation

In Section 6.6 I presented for overview only a part of the full application example and explained my approach based on it. For reference, in Section B.1 I show the full application example on which my work is based. In particular, the full application example contains additional Damage Scenarios, Threat Scenarios, Attack Paths and Attack Feasibility Ratings, and additional Cybersecurity Requirements.

In Section B.2 I present a sample of the digital signage systems (DSS) I found. Here I name the installation site, show by which physical access mechanism these are protected and how these DSS can be unlocked.

B.1 Complete application example

B.1.1 Phase 1: System analysis at environment level

B.1.1.1 [WP-09-01] Item definition at system environment level

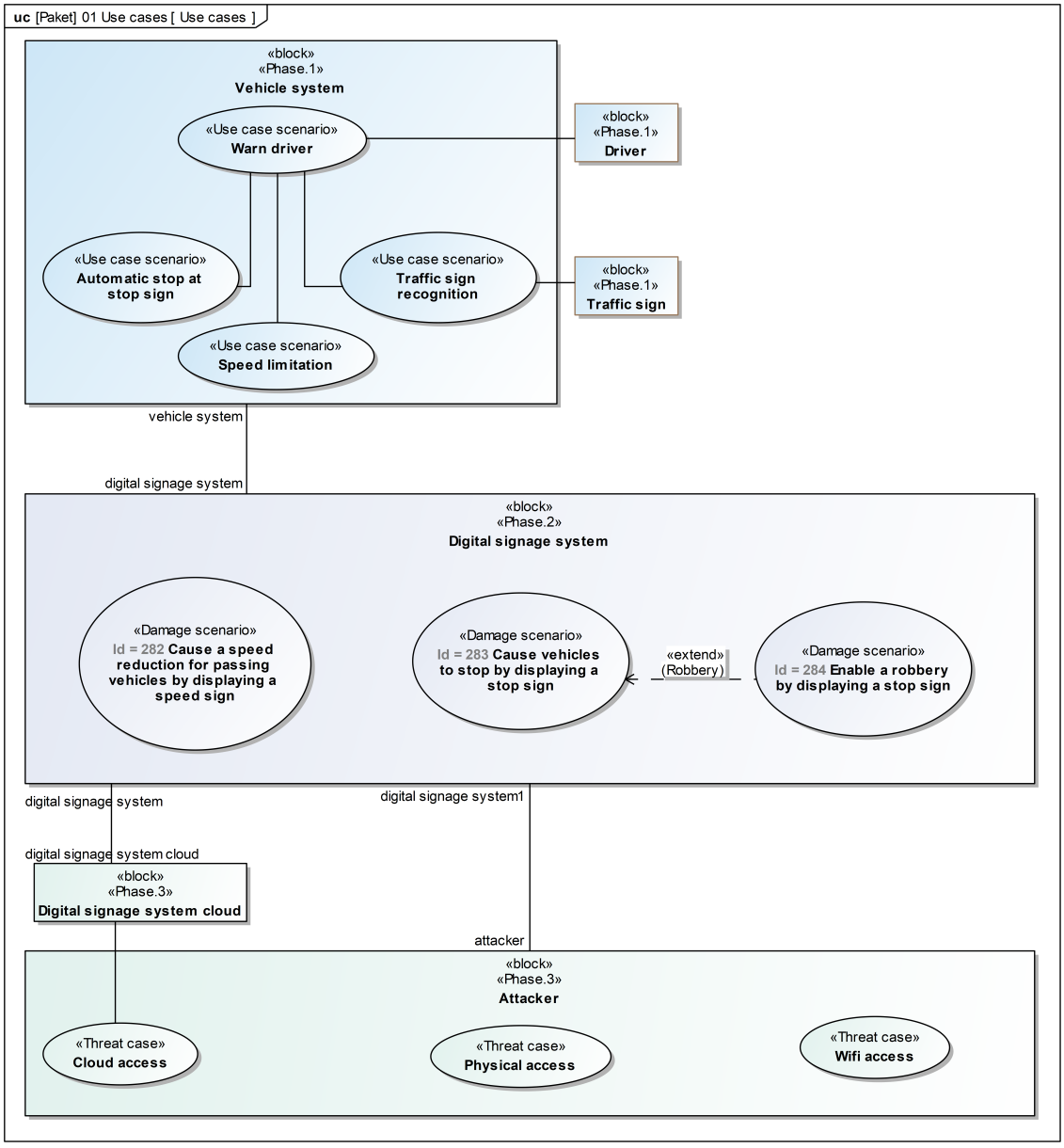


Figure A-7: Use cases, damage scenarios and threat scenarios at the system environment level (related to [WP-09-01], [WP-15-01], [WP-15-03]).

#	Name	Requirement description	Traced To
1	122 Active traffic sign assistant	The system shall actively support the driver in recognizing traffic signs and assist the driver in the prevention of damage.	Visualization: Traffic sign recognition
2	122.1 Traffic sign recognition	The system shall support the driver in traffic sign recognition	Traffic sign recognition
3	122.1.2 Speed sign recognition	The system shall inform the driver about detected speed signs.	
4	122.1.3 Stop sign recognition	The system shall inform the driver about detected stop signs	
5	122.2 Reaction to recognized traffic signs	The system shall warn the driver and actively assist in the prevention of damage.	
6	122.2.1 Driver warning	The system must be able to warn the driver.	Warn driver
7	122.2.1.1 Speed sign warning	The system shall warn the driver when the detected maximum speed is exceeded.	
8	122.2.1.2 Stop sign warning	The system shall warn the driver if the driver does not stop before a stop sign detected by the system.	
9	122.2.2 Driver assistance	The driver shall be actively supported by the system in reacting to recognized traffic signs.	Speed limitation Automatic stop at stop sign
10	122.2.2.1 Speed assistance	The system shall make it possible to automatically reduce th speed of the vehicle to the detected maximum speed.	
11	122.2.2.2 Stop sign assistance	The system shall enable the driver's vehicle to stop automatically in front of stop signs.	
12	122.2.2.3 Driver response	The driver must be able to cancel the vehicle assistance at any time.	

Figure A-8: Derived requirements (related to [WP-09-01]).

B.1.2 Phase 2: Impact analysis at environment level

B.1.2.1 [WP-15-01] Damage scenarios

#	Name	Damage scenario description	Traced To
1	282 Cause a speed reduction for passing vehicles by displaying a speed sign	The digital signage system is located (DSS) on the right side of a road in the city. The maximum permitted speed is 70 km/h. Vehicle A and vehicle B are driving at 70 km/h on the road. Vehicle B has a small distance to vehicle A. Vehicle A has sign recognition and speed reduction turned on. An attacker hacks the DSS and uploads a 20 km/h sign to the DSS. As a result, vehicle A abruptly reduces its speed to 20 km/h. Vehicle B crashes into vehicle A.	Visualization - Damage scenario - Traffic sign recognition
2	283 Cause vehicles to stop by displaying a stop sign	Same as DS.1, but attacker uploads a stop sign and Vehicle A has active stop sign assistant (like Tesla Model 3, USA, Beta, Full Service Driving - FSD))	
3	284 Enable a robbery by displaying a stop sign	Same as DS.2, but vehicle stop is used for robbery.	

Figure A-9: Description of a damage scenario as part of [WP-15-01].

B.1.2.2 [WP-15-02] Assets with cybersecurity properties

#	Name	Asset	Cybersecurity property
1	282 Cause a speed reduction for passing vehicles by displaying a speed sign	Camera sensor unit	Integrity
2	283 Cause vehicles to stop by displaying a stop sign	Camera sensor unit	Integrity
3	284 Enable a robbery by displaying a stop sign	Camera sensor unit	Integrity

Figure A-10: Determination of affected assets and compromised cybersecurity properties of a damage scenario, as part of [WP-15-02].

B.1.2.3 [WP-15-04] Impact ratings with associated impact categories

#	Name	Safety impact - Exposure	Safety impact - Controllability	Safety impact - ASIL	Safety impact - Severity	Financial impact	Operational impact	Privacy impact
1	282 Cause a speed reduction for passing vehicles by displaying a speed sign	Low	Normal	QM	Major	Major	Major	Negligible
2	283 Cause vehicles to stop by displaying a stop sign	Low	Difficult/Uncontrollable	QM	Severe	Severe	Severe	Negligible
3	284 Enable a robbery by displaying a stop sign	Very low	Normal	QM	Severe	Severe	Severe	Negligible

Figure A-11: Impact ratings for safety, financial, operational and privacy as part of [WP-15-03].

B.1.3 Phase 3: Security analysis at environment level

B.1.3.1 [WP-15-03] Threat scenarios

#	Name	Threat scenario description	Affected damage scenario
1	299 Cloud access	Attack on the cloud system via brute-force attack on the login page	282 Cause a speed reduction for passing vehicles by displaying a speed sign 283 Cause vehicles to stop by displaying a stop sign 284 Enable a robbery by displaying a stop sign
2	300 Physical access	Open case of digital signage system and get access to computer system	282 Cause a speed reduction for passing vehicles by displaying a speed sign 283 Cause vehicles to stop by displaying a stop sign 284 Enable a robbery by displaying a stop sign
3	301 Wifi access	Man-in-the-middle attack via Wifi	282 Cause a speed reduction for passing vehicles by displaying a speed sign 283 Cause vehicles to stop by displaying a stop sign 284 Enable a robbery by displaying a stop sign

Figure A-12: Description of a threat scenario in the context of [WP-15-03].

#	Name	Threat scenario description	Related threats from UN R155
1	299 Cloud access	Attack on the cloud system via brute-force attack on the login page	222.3.1 (28.1) Software bugs. The presence of software bugs can be a basis for potential exploitable vulnerabilities. This is particularly true if software has not been tested to verify that known bad code/bugs is not present and reduce the risk of unknown bad code/bugs being present 220.3.1 (18.1) External interfaces such as USB or other ports used as a point of attack, for example through code injection
2	300 Physical access	Open case of digital signage system and get access to computer system	220.3.1 (18.1) External interfaces such as USB or other ports used as a point of attack, for example through code injection 222.3.1 (28.1) Software bugs. The presence of software bugs can be a basis for potential exploitable vulnerabilities. This is particularly true if software has not been tested to verify that known bad code/bugs is not present and reduce the risk of unknown bad code/bugs being present
3	301 Wifi access	Man-in-the-middle attack via Wifi	220.3.1 (18.1) External interfaces such as USB or other ports used as a point of attack, for example through code injection 222.3.1 (28.1) Software bugs. The presence of software bugs can be a basis for potential exploitable vulnerabilities. This is particularly true if software has not been tested to verify that known bad code/bugs is not present and reduce the risk of unknown bad code/bugs being present

Figure A-13: Description of a threat scenario in the context of [WP-15-03] and referencing to UN R155 threats.

B.1.4 Phase 4: Analysis at system level

B.1.4.1 [WP-09-01] Item definition at system level

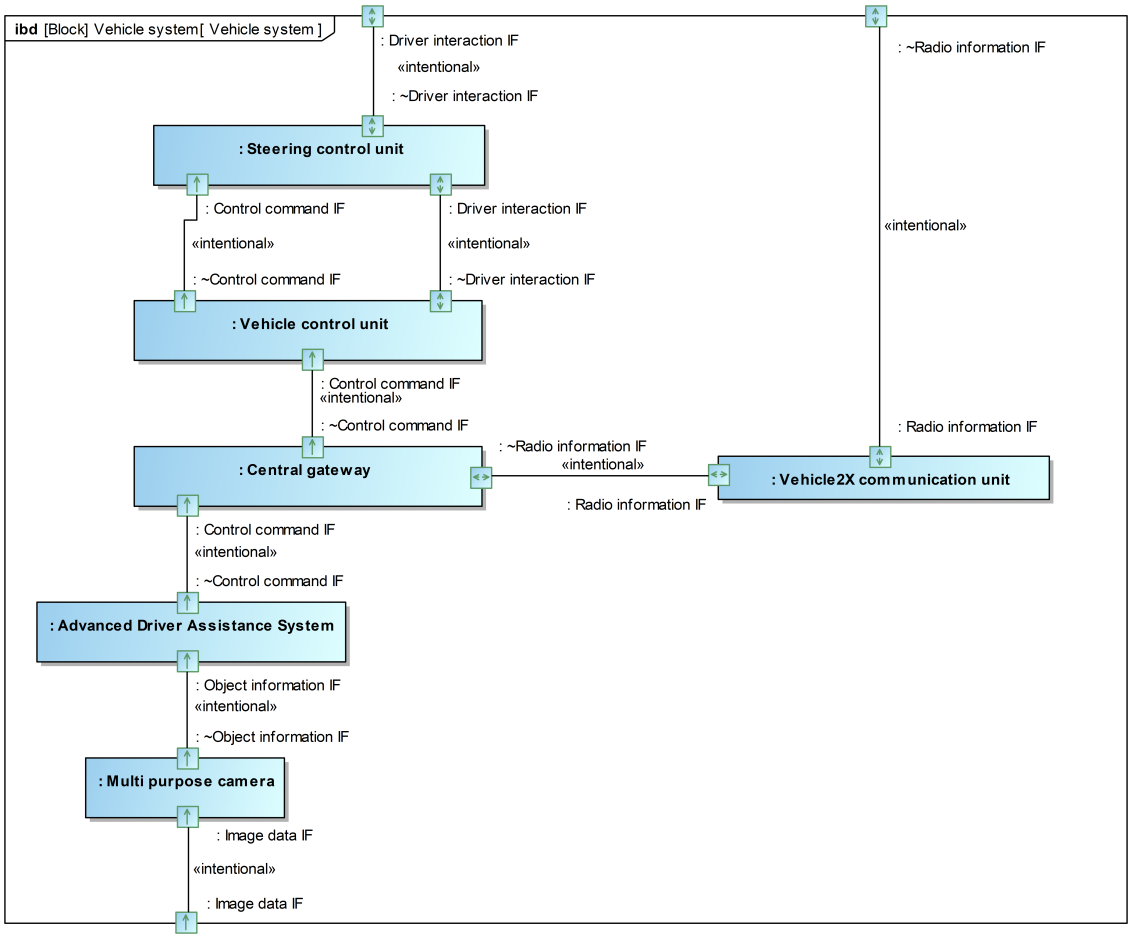


Figure A-14: Identification of necessary components and component relationships for the realisation of the considered use cases as part of [WP-09-01].

#	Name	Requirement description.	Traced To
1	292 Sign recognition	The vehicle must dedect traffic signs with the help of a camera sensor.	Use cases Vehicle system
2	292.1 Processing of object information	The camera sensor must send information about detected traffic signs to the Advanced Driver Assistance System so that the Advanced Driver Assistance System can derive a decision.	
3	292.1.1 Forwarding ADAS decision	The decision of the ADAS must be forwarded via the Central Gateway, the Vehicle Control System up to the Steering Control System.	
4	292.2 Driver warning through Steering Control	If the Steering Control receives a message from the ADAS about the Vehicle Control, the driver must be warned.	
5	292.2.1 Reaction to driver warning	If the driver confirms the warning or does not react, the Steering Control must inform the Vehicle Control. If not, it must not.	
6	292.3 Adjust speed	If the Steering Control informs the Vehicle Control that the driver has acknowledged or ignored the warning and thus has not discarded it, the Vehicle Control must adjust the speed.	

Figure A-15: Refinement of the requirements from Phase 1 from a white box perspective (related to [WP-09-01]).

B.1.5 Phase 5: Security analysis at environment level

B.1.5.1 [WP-15-05] Attack paths

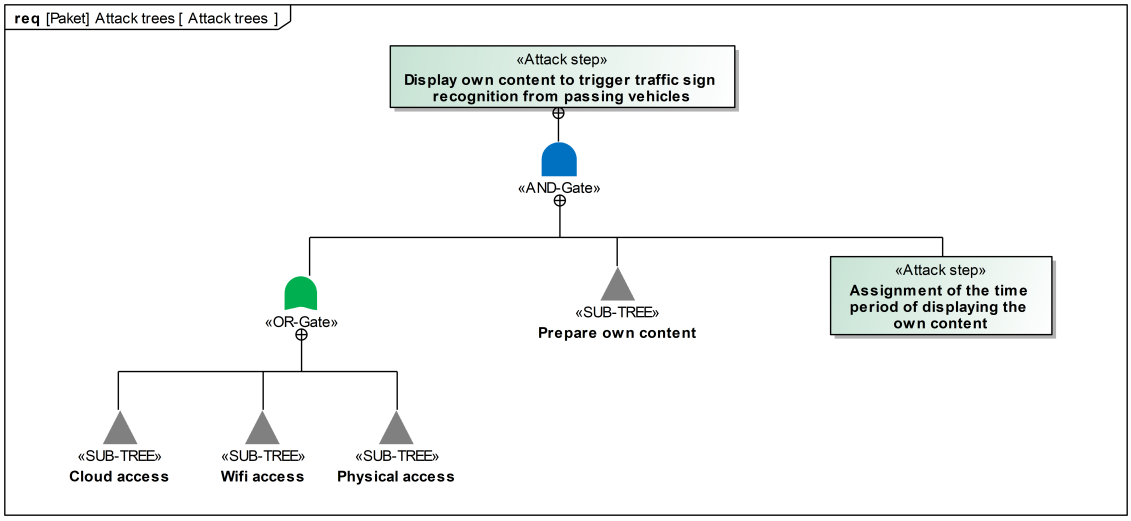


Figure A-16: Overall attack tree, for tricking the object recognition of a vehicle by a digital signage system (related to [WP-15-05]).

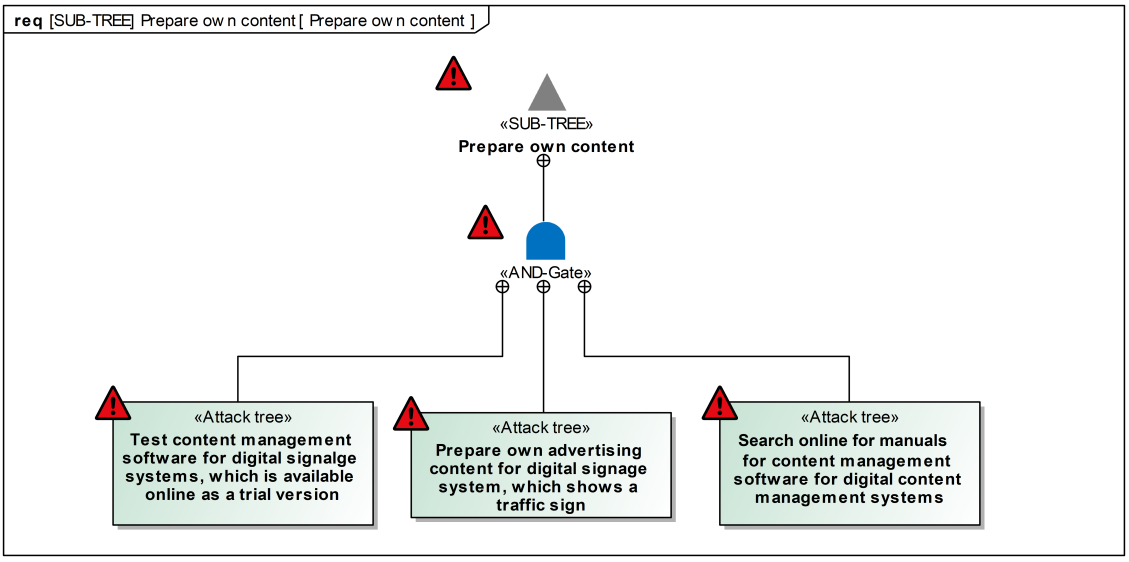


Figure A-17: Prepare own content attack tree.

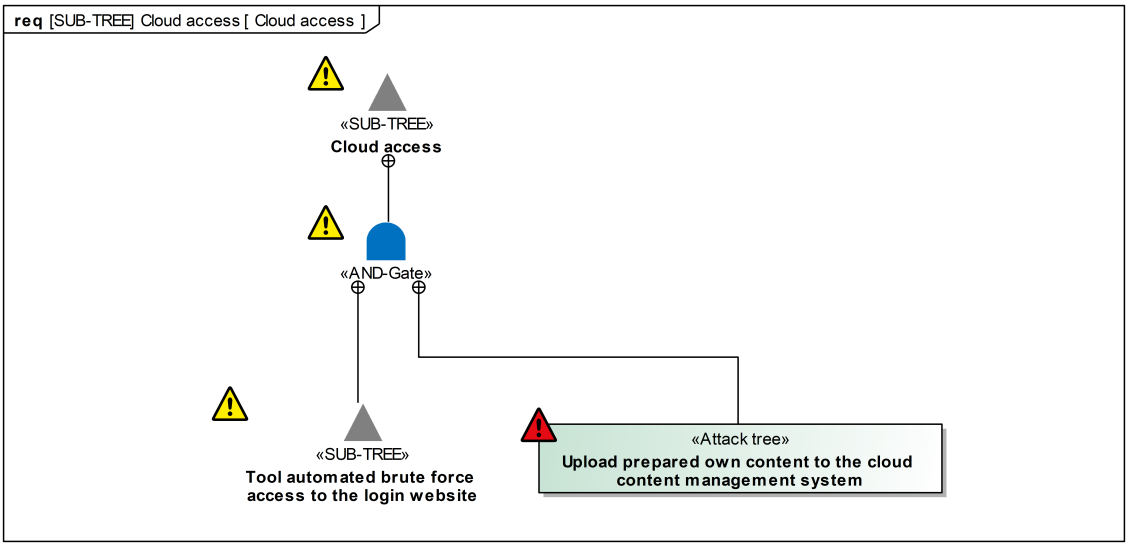


Figure A-18: Cloud access attack tree.

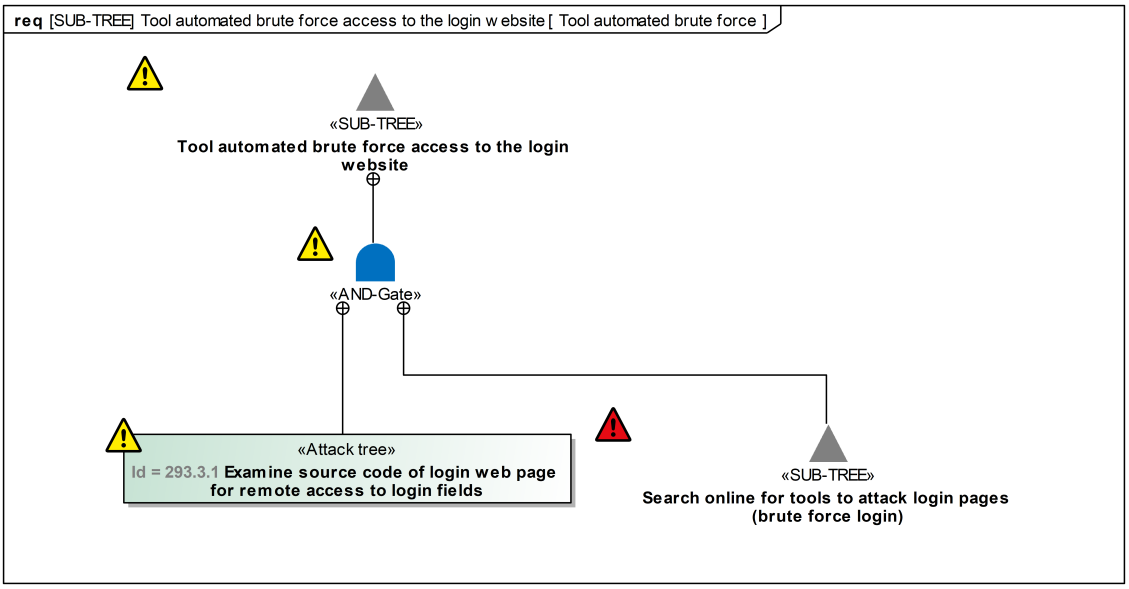


Figure A-19: Tool automated brute force attack tree.

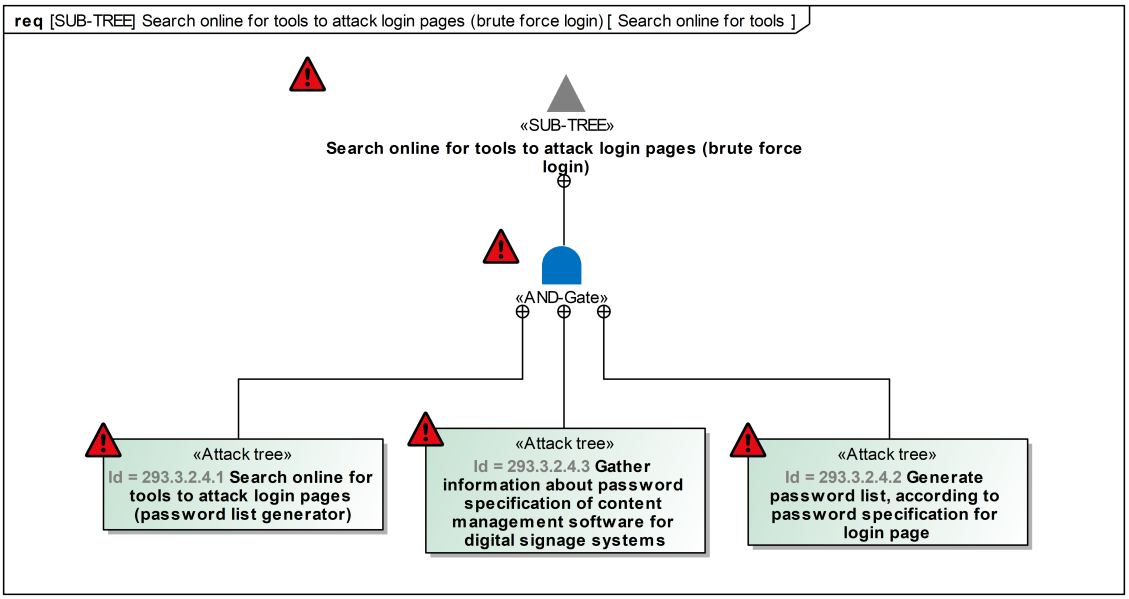


Figure A-20: Search online for tools attack tree.

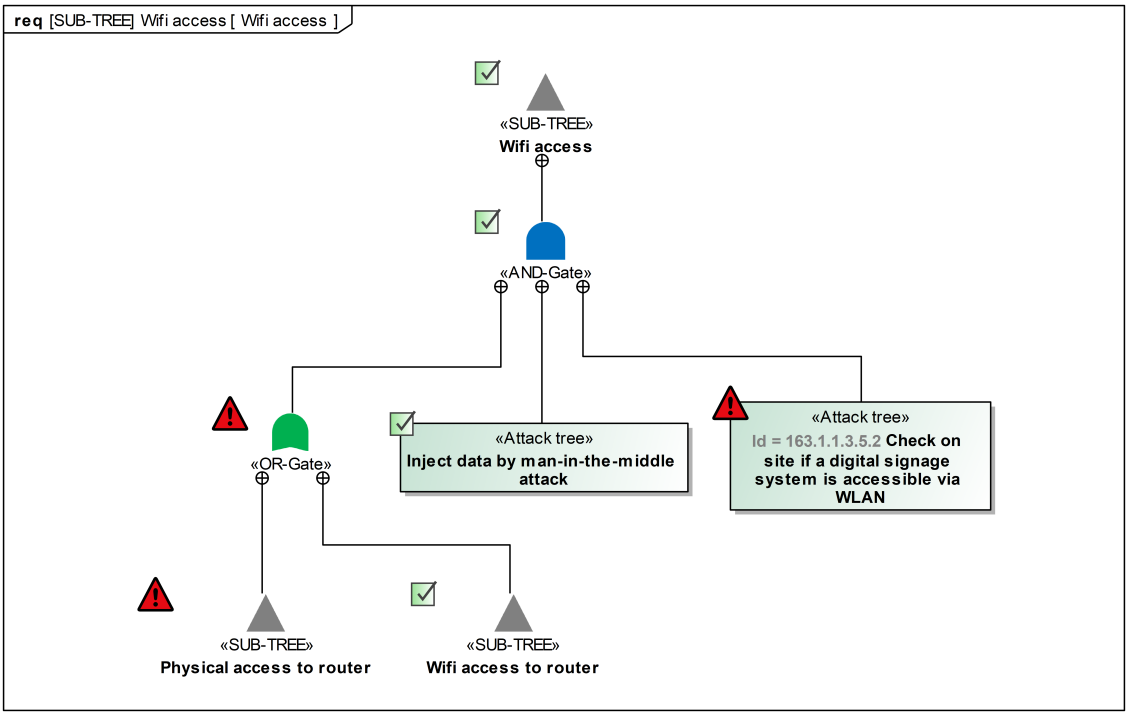


Figure A-21: Wifi access attack tree.

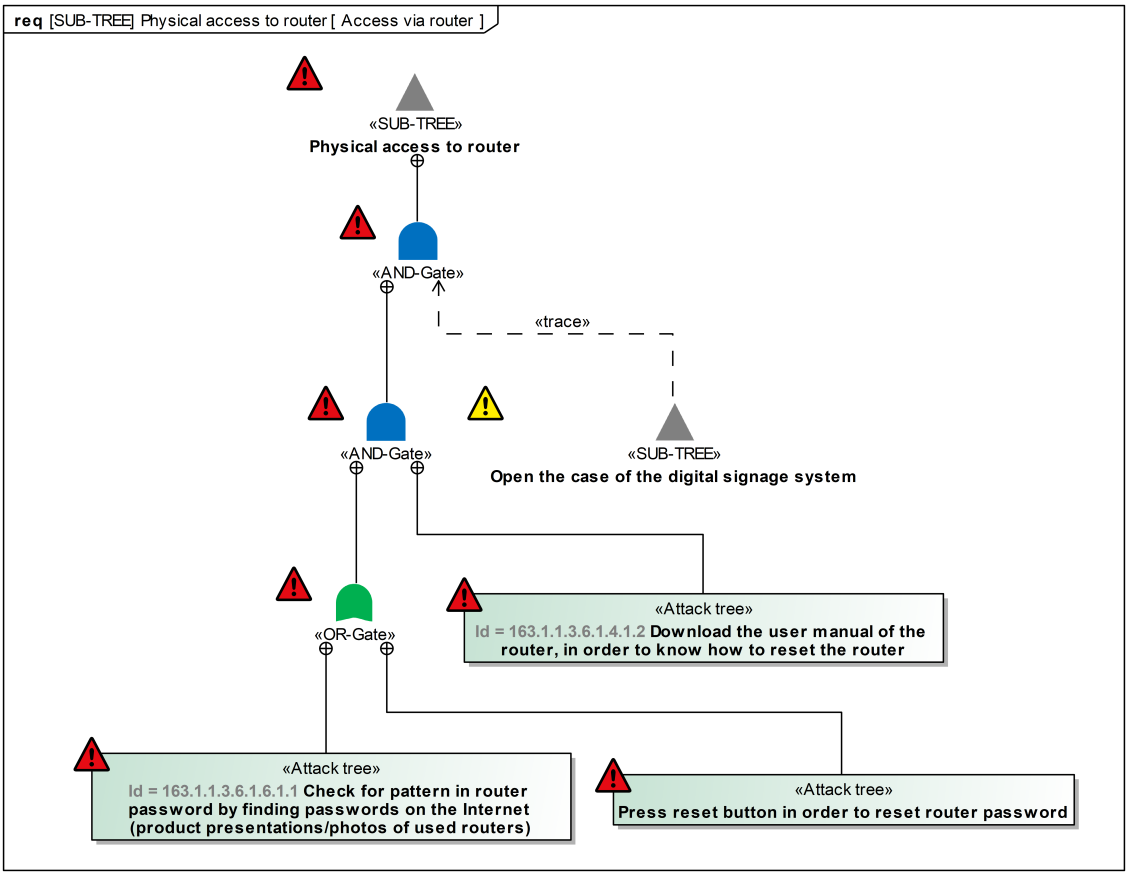


Figure A-22: Access via router attack tree.

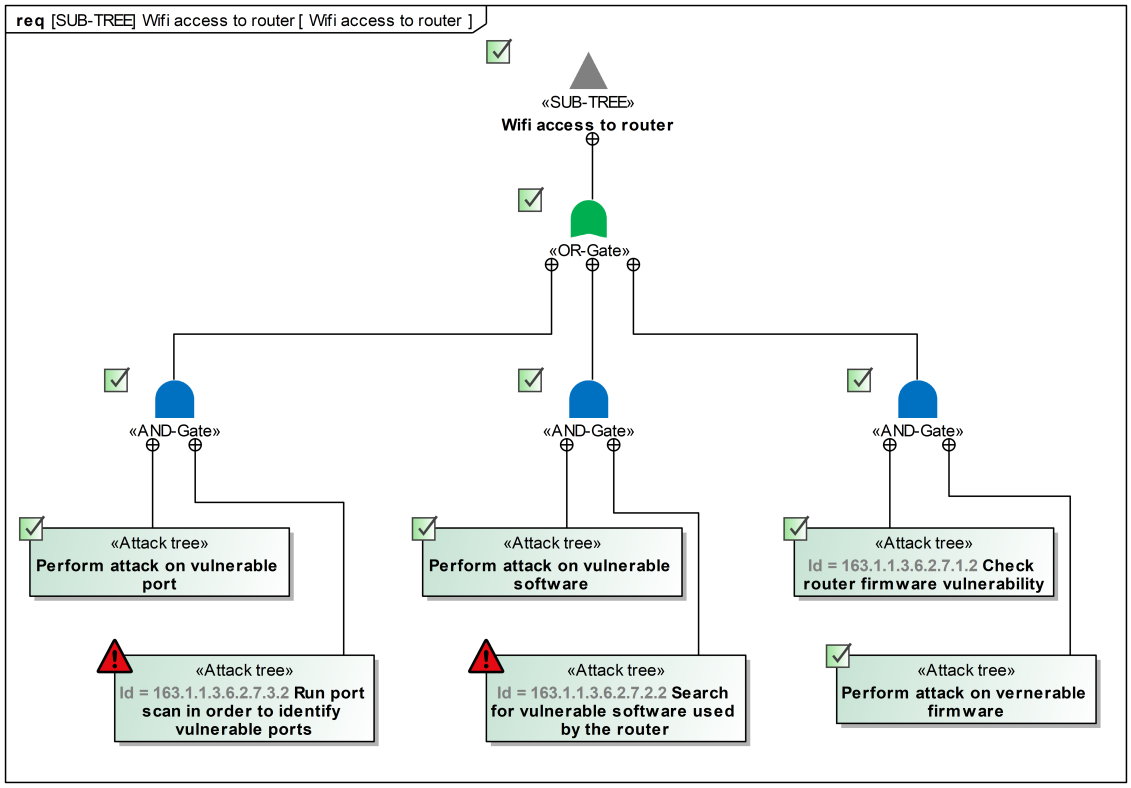


Figure A-23: Wifi access to router attack tree.

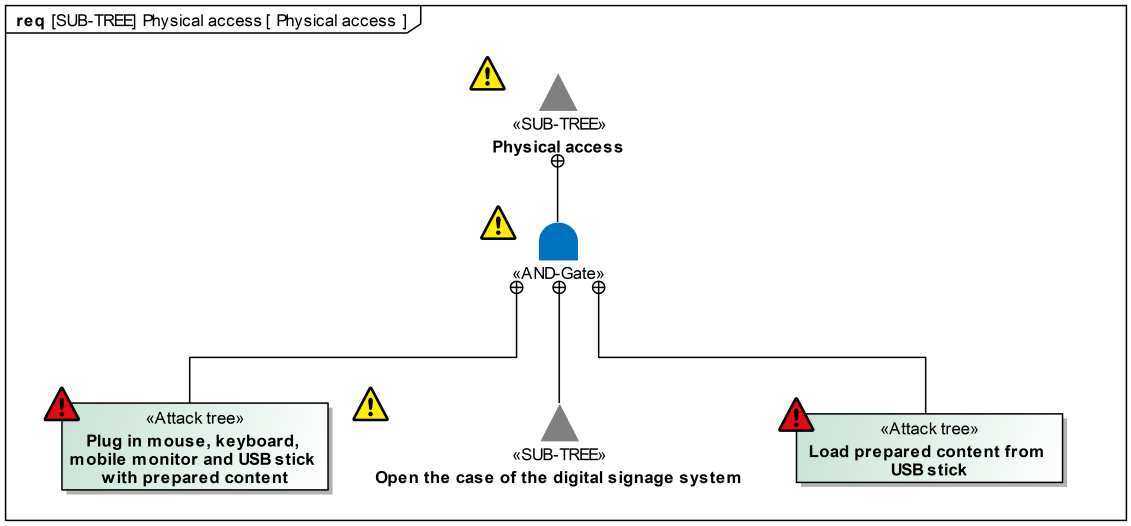


Figure A-24: Physical access attack tree.

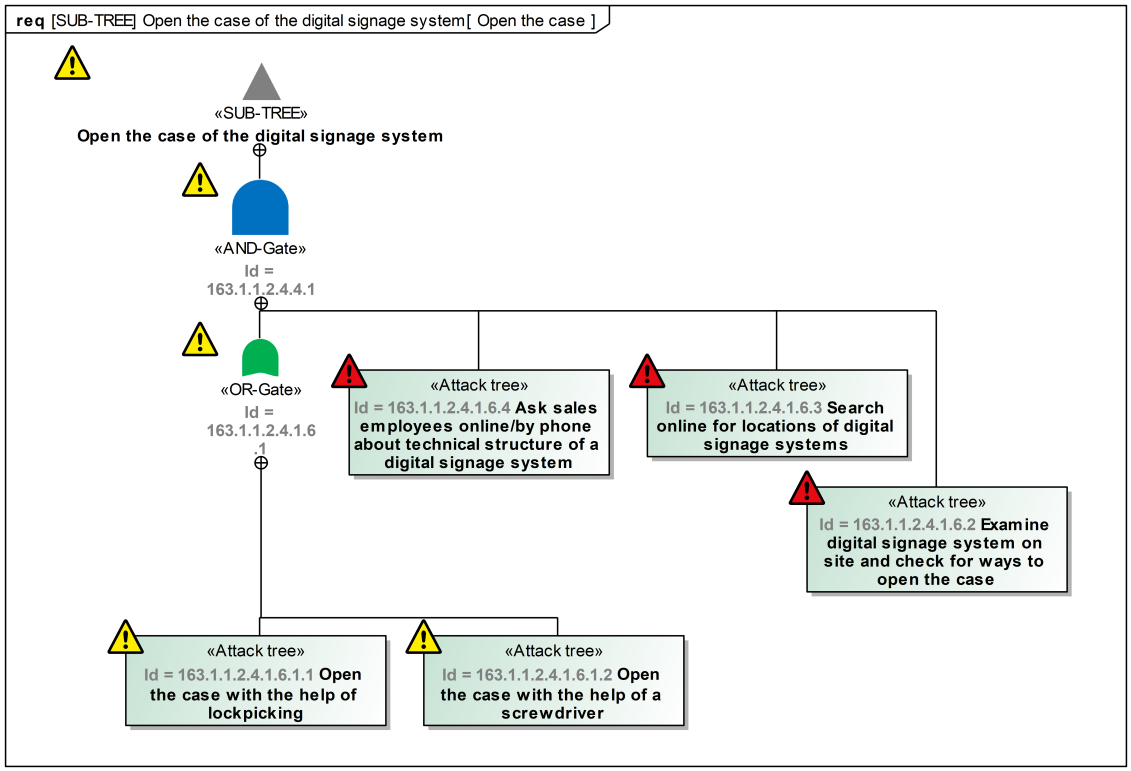


Figure A-25: Open the case attack tree.

B.1.5.2 [WP-15-06] Attack feasibility ratings

#	Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163 Display own content to trigger traffic sign recognition from passing vehicles						Medium
2	163.1						Medium
3	163.1.1						Medium
4	163.1.1.3 Wifi access						Very low
27	163.1.1.2 Physical access						Medium
39	163.1.1.1 Cloud access						Medium
50	163.1.2 Prepare own content						High
55	163.1.3 Assignment of the time period of displaying the own content	Layman	Restricted	Specialized	<= 1 day	Easy	High

Figure A-26: Overall attack feasibility rating.

#	Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.2 Prepare own content						High
2	163.1.2.1						High
3	163.1.2.1.1 Search online for manuals for content management software for digital content management systems	Layman	Public	Standard	<= 1 day	Unlimited	High
4	163.1.2.1.2 Test content management software for digital signalge systems, which is available online as a trial version	Proficient	Public	Specialized	<= 1 week	Easy	High
5	163.1.2.1.3 Prepare own advertising content for digital signage system, which shows a traffic sign	Layman	Public	Specialized	<= 1 day	Moderate	High

Figure A-27: Prepare own content attack feasibility rating.

#	Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.1 Cloud access						Medium
2	163.1.1.1.2						Medium
3	163.1.1.1.2.1 Tool automated brute force access to the login website						Medium
4	163.1.1.1.2.1.1						Medium
5	293.3.1 Examine source code of login web page for remote access to login fields	Expert	Restricted	Specialized	<= 1 week	Unlimited	Medium
6	293.3.2 Search online for tools to attack login pages (brute force login)						High
7	293.3.2.4						High
8	293.3.2.4.1 Search online for tools to attack login pages (password list generator)	Proficient	Public	Specialized	<= 1 week	Unlimited	High
9	293.3.2.4.2 Generate password list, according to password specification for login page	Proficient	Public	Specialized	<= 1 week	Unlimited	High
10	293.3.2.4.3 Gather information about password specification of content management software for digital signage systems	Layman	Public	Standard	<= 1 week	Unlimited	High
11	163.1.1.1.2.2 Upload prepared own content to the cloud content management system	Layman	Public	Specialized	<= 1 day	Unlimited	High

Figure A-28: Attack feasibility rating for the attack via cloud access (related to [WP-15-06]).

#	△ Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.1.2.1 Tool automated brute force access to the login website						Medium
2	163.1.1.1.2.1.1						Medium
3	293.3.1 Examine source code of login web page for remote access to login fields	Expert	Restricted	Specialized	<= 1 week	Unlimited	Medium
4	293.3.2 Search online for tools to attack login pages (brute force login)						High

Figure A-29: Tool automated brute force attack feasibility rating.

#	Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	293.3.2 Search online for tools to attack login pages (brute force login)						High
2	293.3.2.4						High
3	293.3.2.4.3 Gather information about password specification of content management software for digital signage systems	Layman	Public	Standard	<= 1 week	Unlimited	High
4	293.3.2.4.1 Search online for tools to attack login pages (password list generator)	Proficient	Public	Specialized	<= 1 week	Unlimited	High
5	293.3.2.4.2 Generate password list, according to password specification for login page	Proficient	Public	Specialized	<= 1 week	Unlimited	High

Figure A-30: Search online for tools attack feasibility rating.

#	△ Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.3 Wifi access						Very low
2	163.1.1.3.5						Very low
3	163.1.1.3.5.1						High
4	163.1.1.3.5.1.1 Physical access to router						High
11	163.1.1.3.5.1.2 Wifi access to router						Very low
22	163.1.1.3.5.2 Check on site if a digital signage system is accessible via WLAN	Layman	Public	Standard	<= 1 day	Easy	High
23	163.1.1.3.5.3 Inject data by man-in-the-middle attack	Expert	Restricted	Bespoke	<= 1 month	Difficult/none	Very low

Figure A-31: Wifi access attack feasibility rating.

#	△ Name		Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.3.5.1.1 Physical access to router							High
2	163.1.1.3.6.1.4							High
3	163.1.1.3.6.1.4.1							High
4	163.1.1.3.6.1.4.1.1							High
5	163.1.1.3.6.1.6.1.1	Check for pattern in router password by finding passwords on the Internet (product presentations/photos of used routers)	Proficient	Confidential	Standard	<= 1 week	Unlimited	High
6	163.1.1.3.6.1.6.1.2	Press reset button in order to reset router password	Layman	Public	Standard	<= 1 day	Unlimited	High
7	163.1.1.3.6.1.4.1.2	Download the user manual of the router, in order to know how to reset the router	Layman	Public	Standard	<= 1 day	Unlimited	High

Figure A-32: Access via router attack feasibility rating.

#	Name		Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.3.5.1.2 Wifi access to router							Very low
2	163.1.1.3.6.2.7							Very low
3	163.1.1.3.6.2.7.1							Very low
4	163.1.1.3.6.2.7.1.2	Check router firmware vulnerability	Expert	Confidential	Bespoke	<= 1 month	Unlimited	Very low
5	163.1.1.3.6.2.7.1.1	Perform attack on vernerable firmware	Expert	Confidential	Bespoke	<= 1 month	Easy	Very low
6	163.1.1.3.6.2.7.2							Very low
7	163.1.1.3.6.2.7.2.2	Search for vulnerable software used by the router	Proficient	Public	Standard	<= 1 day	Unlimited	High
8	163.1.1.3.6.2.7.2.1	Perform attack on vulnerable software	Expert	Confidential	Bespoke	<= 1 month	Easy	Very low
9	163.1.1.3.6.2.7.3							Very low
10	163.1.1.3.6.2.7.3.2	Run port scan in order to identify vulnerable ports	Proficient	Public	Specialized	<= 1 day	Easy	High
11	163.1.1.3.6.2.7.3.1	Perform attack on vulnerable port	Expert	Confidential	Bespoke	<= 1 week	Easy	Very low

Figure A-33: Wifi access to router attack feasibility rating.

#	Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.2 Physical access						Medium
2	163.1.1.2.4						Medium
3	163.1.1.2.4.4 Open the case of the digital signage system						Medium
11	163.1.1.2.4.2 Plug in mouse, keyboard, mobile monitor and USB stick with prepared content	Layman	Public	Specialized	<= 1 day	Moderate	High
12	163.1.1.2.4.3 Load prepared content from USB stick	Layman	Public	Standard	<= 1 day	Moderate	High

Figure A-34: Physical access attack feasibility rating.

#	△ Name	Specialist expertise	Knowledge of the item or component	Equipment	Elapsed time	Window of opportunity	Attack feasibility rating
1	163.1.1.2.4.4 Open the case of the digital signage system						Medium
2	163.1.1.2.4.4.1						Medium
3	163.1.1.2.4.1.6.1						Medium
4	163.1.1.2.4.1.6.1.1 Open the case with the help of lockpicking	Proficient	Public	Specialized	<= 1 day	Difficult/none	Medium
5	163.1.1.2.4.1.6.1.2 Open the case with the help of a screwdriver	Layman	Public	Standard	<= 1 day	Difficult/none	Medium
6	163.1.1.2.4.1.6.2 Examine digital signage system on site and check for ways to open the case	Layman	Public	Standard	<= 1 week	Easy	High
7	163.1.1.2.4.1.6.3 Search online for locations of digital signage systems	Layman	Public	Standard	<= 1 week	Unlimited	High
8	163.1.1.2.4.1.6.4 Ask sales employees online/by phone about technical structure of a digital signage system	Proficient	Public	Standard	<= 1 week	Easy	High

Figure A-35: Open the case of the digital signage system attack feasibility rating.

B.1.5.3 [WP-15-07] Risk values

#	△ Name	Attack tree	Attack tree - Feasibility rating	Affected damage scenario
1	299 Cloud access	163.1.1.1 Cloud access	Medium	282 Cause a speed reduction for passing vehicles by displaying a speed sign 283 Cause vehicles to stop by displaying a stop sign 284 Enable a robbery by displaying a stop sign
2	300 Physical access	163.1.1.2 Physical access	Medium	282 Cause a speed reduction for passing vehicles by displaying a speed sign 283 Cause vehicles to stop by displaying a stop sign 284 Enable a robbery by displaying a stop sign
3	301 Wifi access	163.1.1.3 Wifi access	Very low	282 Cause a speed reduction for passing vehicles by displaying a speed sign 283 Cause vehicles to stop by displaying a stop sign 284 Enable a robbery by displaying a stop sign

Figure A-36: Risk calculation for a damage scenario (related to [WP-15-07]).

B.1.5.4 [WP-15-08] Risk treatment decisions

#	Name	Safety risk level	Financial risk level	Operational risk level	Privacy risk level	Risk treatment option
1	299 Cloud access	4	4	4	1	reduce the risk
2	300 Physical access	4	4	4	1	reduce the risk
3	301 Wifi access	2	2	2	1	share the risk

Figure A-37: Determination of a risk treatment (related to [WP-15-08]).

B.1.5.5 [WP-09-03/04] Cybersecurity goals/claims

#	Name	Asset	Threat scenario	Cybersecurity assurance level/ CAL	Attack tree - Feasibility rating	Required trust level/ RTL
1	196 Ensure the integrity of the traffic sign recognition of speed signs.	Camera sensor unit	299 Cloud access 300 Physical access	CAL4	High	4H
2	202 Ensure the integrity of the traffic sign recognition of stop signs.	Camera sensor unit	299 Cloud access 300 Physical access	CAL4	High	4H

Figure A-38: Derivation of a cybersecurity goal (related to [WP-09-03]).

#	Name	Asset	Threat scenario	Cybersecurity assurance level/ CAL	Attack tree - Feasibility rating	Required trust level/ RTL
1	297 The risk is transferred to insurance. The feasibility of a WiFi attack is very low.	Camera sensor unit	301 Wifi access	CAL4	Very low	4VL

Figure A-39: Derivation of a cybersecurity claims (related to [WP-09-04]).

B.1.5.6 [WP-09-05] Verification report for cybersecurity goals

#	Work product	Consistency	Correctness	Completness	Comment
1	[WP-09-01] Item definition - Environment level	Not demanded	Satisfied	Satisfied	
2	[WP-09-01] Item definition - Vehicle level	Not demanded	Satisfied	Satisfied	
3	[WP-15-01] Damage scenarios	Not demanded	Satisfied	Partially satisfied	Further damage scenarios possible
4	[WP-15-02] Assets with cybersecurity properties	Not demanded	Satisfied	Satisfied	
5	[WP-15-03] Threat scenarios	Not demanded	Satisfied	Partially satisfied	Further threat scenarios possible
6	[WP-15-04] Impact ratings with associated impact categories	Not demanded	Satisfied	Satisfied	
7	[WP-15-05] Attack paths	Not demanded	Satisfied	Satisfied	
8	[WP-15-06] Attack feasibility ratings	Not demanded	Satisfied	Satisfied	
9	[WP-15-07] Risk values	Not demanded	Satisfied	Satisfied	
10	[WP-15-08] Risk treatment decisions	Satisfied	Satisfied	Satisfied	
11	[WP-09-03] Cybersecurity goals	Satisfied	Satisfied	Satisfied	
12	[WP-09-04] Cybersecurity claims	Satisfied	Satisfied	Satisfied	

Figure A-40: Verification report for cybersecurity goals [WP-09-05].

B.1.5.7 [WP-09-06] Cybersecurity concept

#	Text	Allocation	Mitigation	Security design pattern
1	When detecting a speed sign using a Multi purpose camera, the context of the speed sign has to be associated using the Advanced Driver Assistance System.	: Multi purpose camera : Advanced Driver Assistance System	238 (M22) Secure external interfaces	281.1 (01) Context box pattern
2	When detecting a stop sign using a Multi purpose camera, the context of the stop sign has to be associated using the Advanced Driver Assistance System.	: Multi purpose camera : Advanced Driver Assistance System	238 (M22) Secure external interfaces	281.1 (01) Context box pattern
3	The Advanced Driver Assistance System has to support sensor fusion. For this purpose, the image recognition of speed signs has to be supplemented by a Multi purpose camera and depth information by a Radar sensor.	: Advanced Driver Assistance System Radar sensor : Multi purpose camera	223 (M10) Message verification	280.10 (10) Sensor fusion
4	The Advanced Driver Assistance System has to support sensor fusion. For this purpose, the image recognition of stop signs has to be supplemented by a Multi purpose camera and depth information by a Radar sensor.	: Advanced Driver Assistance System Radar sensor : Multi purpose camera	223 (M10) Message verification	280.10 (10) Sensor fusion
5	The Advanced Driver Assistance System must log which traffic sign was detected and from which data sources it was detected (multi purpose camera or radar sensor or both sensors).	: Advanced Driver Assistance System Radar sensor	239 (M23) Cybersecurity best practices	278.3.10 (6.11) Security logger and auditor

Figure A-41: Assignment of cybersecurity controls to vehicle components and derivation of cybersecurity requirements (related to [WP-09-06]).

B.1.5.8 [WP-09-07] Verification report of cybersecurity concept

#	Work product	▽ Correctness	Consistency	Completness	Comment
1	[WP-09-03] Cybersecurity goals [WP-09-06] Cybersecurity concept	Satisfied	Satisfied	Partially satisfied	Further damage/threat scenarios possible
2	[WP-09-04] Cybersecurity claims [WP-09-06] Cybersecurity concept	Not demanded	Satisfied	Not demanded	Further damage/threat scenarios possible

Figure A-42: Verification report of cybersecurity concept [WP-09-07].

B.2 Physical access to digital signage systems

In this section I present examples of tools that can be used to physically open a digital signage system (DSS) and on this basis manipulate the displayed content of the DSS. A DSS basically consists of a small computer (with USB and HDMI connection), a screen (with HDMI connection) and a mobile communication unit (e.g. a router with LAN connection).

The overview in Figure A-43 is a supplement to the attack step *Open the case of the digital signage system* in Figure A-25 and to the evaluation of the attack step in Figure A-35. I took the photos myself on site.

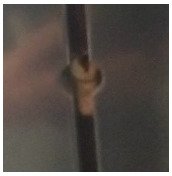
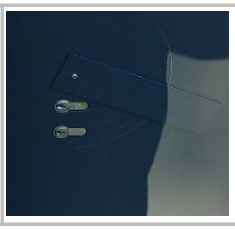
DSS in Paderborn next to a parking space	DSS in Kyoto (Japan) on the street side	DSS in Paderborn on the street side	DSS in Paderborn on the street side
			
Access by torx key	Access by key lock		Access by triangular key
			
Torx key set for 9,66 Euro at Amazon	Lockpicking set for 15,29 Euro at Amazon		Triangular key for 9,00 Euro at Amazon
			

Figure A-43: Overview of access equipment to open a DSS.