Functional Safety: A New Architectural Perspective

Citation for published version (APA):
Khabbaz Saberi, A. (2020). *Functional Safety: A New Architectural Perspective: Model-Based Safety Engineering for Automated Driving Systems*. Eindhoven University of Technology.

Document status and date:
Published: 23/09/2020

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 11. Dec. 2020

Functional Safety

A New Architectural Perspective

Model-Based Safety Engineering for Automated Driving

Arash Khabbaz Saberi

Functional Safety: A New Architectural Perspective

Model-Based Safety Engineering for Automated Driving Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus, prof.dr.ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College voor Promoties,
in het openbaar te verdedigen op maandag 6 april 2020 om 16:00 uur

door

Arash Khabbaz Saberi

geboren te Teheran, Iran

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. J.J. van Wijk
1^e promotor:	prof.dr. M.G.J. van den Brand
2^e promotor:	prof.dr. J.J. Lukkien
leden:	prof.dr. M. Staron (University of Gothenburg)
	prof.dr. P. Zegelaar
	prof.dr. M. Chechik (University of Toronto)
	prof.dr. P. Lago (Vrije Universiteit Amsterdam)
adviseur:	dr. I. Ibarra (Dyson, England)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Functional Safety: A New Architectural Perspective

Model-Based Safety Engineering for Automated Driving Systems

Arash Khabbaz Saberi

The work in this thesis has been carried out by a collaboration between TNO Automotive and Eindhoven University of Technology.

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-94-640-2185-1

Cover design by Negin Bokaeiolmousavi and Forouhar Shakeri
Printed by Gildeprint

Copyright © Arash Khabbaz Saberi, 2020.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronically, mechanically, photocopying, recording or otherwise, without prior permission of the author.

Acknowledgment

More than four years ago, I challenged myself to do a Ph.D. research alongside my job. It seemed to be an easy task back then! I thought how hard can it be? My job at TNO already involves doing research. Time proved me wrong! This challenge was not something that I could have taken alone; luckily I had the help of many amazing people without whom I would not be able to finish this thesis. I enjoyed my time as a Ph.D. candidate and learned many new lessons and skills that surely I will take with me through the rest of my life. I want to express my gratitude to everyone who was involved with my projects.

First and foremost, my special gratitude and appreciation goes to my supervisors at the university, Mark van den Brand and Johan Lukkien who believed in me and gave me the chance to try. Thanks for all your guidance, both technicals and non-technical. From TNO side, my managers Daan de Cloe, Bastiaan Krosse, and Maurice Kwakkernaat, who supported me through thick and thin. I value our time together greatly, and thanks for providing me the support and guidance to go through this challenge.

I had the luxury of having many advisors during these years who helped me in different ways. I thank you all, without your advise I would not be able to be where I am today. Special thanks goes to Peter Heuberger, Peter Zomer, Joelle van den Broak, Joke Welten, Mohsen Alirezaei, Yanja Dajsuren, Yaping (Luna) Luo, Ellen van Nunen, Robert Deckers, Henk Goossens, Hala Elrofai, Ion Barosan, and Terry Fruehling for their invaluable advise at the times of need. I want to thank my steering committee who guided me technically during different stages of my Ph.D: Tjerk Bijlsma, Robert Verschuren, Johan van der Kamp, and Frank Benders. My team leads have my respect and thanks: Robert Verschuren and Antoine from the VDEC days, John Vissers and Ron Wouters during my time in OFS, and Jeroen Uittenbogaard during my time in AMT.

I had the honor to collaborate with many, resulting to a few coauthored publications. Besides my supervisors, I want to thank my coauthors Filip Pawel Cichosz, Yaping (Luna) Luo, Sven Jansen, Tjerk Bijlsma, Ellen van Nunen, Francesco Esposto, Jan-Pieter Paardekooper, Frank Benders, Richard Koch, Eric Barbier, John Vissers, Erwin de Gelder, Andre Smulders, Dennis van den Brand, Jos Hegge, Terry Fruehling, Jan Friso Groote, Robert Deckers, Hala Elofai, Olaf op den Camp, and Jeroen Ploeg.

During the past few years, I worked in many projects at TNO. I want to express my gratitude specially to the members of EcoTwin, Streetwise, Simplexity and Accelerate projects. Special thanks goes to Frank Benders, Ron Wouters, Mohsen Alirezaei, Tjerk Bijlsma, Dimitar Binev, Geert Verhaeg, Emilia Silvas, Sytze Kalisvaart, Hala Elrofai, Maurice Kwakkernaat, Henk Goossens, Dennis van den Brand, Denis Pogosov, Sjef Montfort, and Iuliana Dragomir.

I want to thank all the members of SET groups, I had a lot of fun with our monthly meetings as well as gatherings. Special thanks goes to Ana Maria Sutii and Weslley Silva Torres for all your efforts in organizing everything. I want to also thank Omar Alduhaibi, Kousar Aslam, Onder Babur, Miguel Botto Tobar, Felipe Ebert, Priyanka Karkhanis, Thomas Neele, Sangeeth Kochanthara, Fei Yang, Mahmoud Talebi, Mahdi Saeedi Nikoo, Alexander Fedotov, and Rick Erkens. I want to also thank my roommates for all the good times: Raquel Alvarez-Ramirez, Yuexu Chen, Josh Mengerink, Sander de Putter, Rick Erkens, Lars van den Haak, and Muhammad Muhammad.

During my Ph.D., I had the chance to coach a few student projects. I want to thank the ASD/MSD and ST PDEng programs and SET groups for giving me the opportunities. I want to thank Ka ka Tam, Mananpreet Singh, Alexandre Carey, Munkhdalai Sainbileg, Abhimanyu Badrinarayan, Luigi Altamirano Mollo, and Pepijn Hockers. I want to specially thank Dennis van den Brand for his good work and his continues support and contributions.

Also, I want to extend my sincere gratitude to the members of my promotion committee, thanks for your valuable feedback and comments: Miroslaw Staron, Peter Zegelaar, Marsha Chechik, Patricia Lago, and Ireri Ibarra. Special thanks goes to Henk Nijmeijer for his last minute response.

I want to thank all my friends who supported me through out this period. Special thanks to Forouhar and my amazing girlfriend Negin for their work and contribution to the cover page design.

Finally, I want to deepest and most sincere gratitude to my parents, Hadi and Mandana, and my family specially my grandma Soraya and lovely girlfriend Negin, who supported me throughout the years with everything that lead to this moment. Thanks for your love and support.

Arash Khabbaz Saberi
Helmond, March 2020

Summary

The automotive industry has seen a rapid change in the technologies used inside the vehicles. Since the introduction of the first electronic control unit, the impact of electronics and computer science on the quality of the vehicles is increasing every year. Arguably, safety is one of the most important quality attributes of a vehicle that needs special attention in all the stages of the life cycle of a vehicle. The overall safety of a vehicle can be seen from multiple aspects, such as passive safety, active safety, functional safety, etc. Functional safety addresses the hazards that are caused by the malfunctioning of Electrical and/or Electronic (E/E) systems. Many factors impact functional safety such as the organization and management, the development process, the design of the systems, the system type and technologies used in it, the quality control methods, etc. The ISO 26262 standard provides the state of the art of functional safety in the automotive industry with respect to development processes, design principles and safety analysis. A technical committee of subject matter experts from industry defines the content of this standard. The difference in viewpoints, choice of language and industry agenda result in unavoidable (even though slight) inconsistencies in the ISO text. Besides, ensuring unique interpretation of standards by experts is impossible. In this research, we work on the different aspects of functional safety: we study the use of models in performing safety engineering and propose a domain model and SW tooling for modeling, we study the impact of functional safety on architectural patterns and propose a new pattern for safety-critical systems. Furthermore, we investigate the impact of applying ISO 26262 to systems of systems and propose a tailored safety lifecycle based on guidelines of ISO 26262 that is augmented to encompass additional considerations pertinent to systems of systems. Finally, we study the human aspects of the development of safety-critical systems in an R&D environment. We introduced a method for measuring the safety culture in accordance with ISO 26262. This research is done at the Integrated Vehicle Safety (IVS) department of TNO. IVS is active in research related to automated driving with special attention to connected and cooperative mobility. The technology roadmap of IVS is defined around Cooperative Automated Driving (CAD), and functional safety topics are an essential part of this roadmap. It should be mentioned that all the results from this research have been applied in one or more running projects at IVS.

Table of Contents

List of Acronyms

AD	Automated Driving
ASIL	Automotive Safety Integrity Level
E/E	Electrical and/or Electronic
FMEA	Failure Mode and Effects Analysis
FSC	Functional Safety Concept
FSR	Functional Safety Requirements
FTA	Fault Tree Analysis
HARA	Hazard Analysis and Risk Assessment
ICT	Information and Communication Technology
OEM	Original Equipment Manufacturer
TSR	Technical Safety Requirements
UML	Unified Modeling Language
V&V	Verification and Validation

Chapter 1

Introduction

In this thesis, we present our work on model-based approach for safety engineering in the automotive domain. We especially focus on the impact of safety on the architecture design of an automated driving system. In this introduction, we position this work in the system architecture and safety engineering domains. We further present arguments on why this research is relevant.

We start by describing some of the trends in the automotive industry and discuss the impact of these trends on achieving safety. We then discuss the research questions in this thesis in response to the identified challenges. We then explain the context in which this work is created. Finally, we give the outline of this thesis and give some suggestions for reading it.

1.1 Rising Complexity: An Automotive Trend

Electrical and/or Electronic (E/E) systems in combination with software, or in other words *embedded systems*, within a vehicle have been growing in number and complexity. This complexity have been increasing as these systems replace more mechanical systems by introduction of various X-by-wire systems. The introduction of automated driving and smart mobility [24], [88] has accelerated the rise in complexity of the automotive embedded systems as well as their safety criticality. As these systems take more responsibility for the dynamic driving tasks and monitoring the system and the environment, therefore gradually replace the driver, their quality attributes such as safety and security becomes more critical. Also, the race towards Mobility as a Service (MaaS) [36], [89] connects the vehicles to untraditional ecosystem of Information and Communication Technology (ICT) systems and adds a level in the system hierarchy. Evidence of the rise of complexity is the gradual shift towards modern communication networks in the automotive E/E architectures to manage the increasing dependencies and communication load. A timeline of the increase in system complexity and its impact on the system architecture is shown in Figure 1.1.

There are many social and economic motivations for automated driving and smart-mobility including emissions reduction, comfort, efficiency, and road safety. Better road safety is regarded as one of the main motivations of automated driving applications since human errors are currently accounted for more than 90% of road accidents [109]. Road safety trends show improvement over the past several decades, as shown in Figure 1.2. The current road safety relies heavily on having the safety of the infrastructure and ensuring responsible human drivers. By automating the driving task, and therefore removing the human error factor, it is expected to achieve higher levels of road safety. However, transferring the driving tasks from the driver to (in-vehicle) systems means that the safety responsibility is also shifted to (in-vehicle) systems. This shift of responsibility means that the accounts for the road accidents are soon going to change, in that case, how can we categorize if the root cause was a fault in the system (e.g. a software bug) or was it a design mistake?

We view safety of automotive systems from two perspectives: product safety and func-

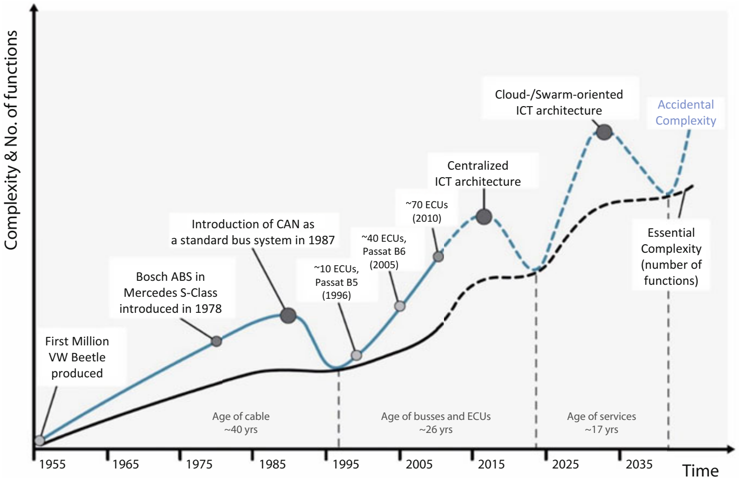


Figure 1.1: The trend of complexity of automotive systems [69]

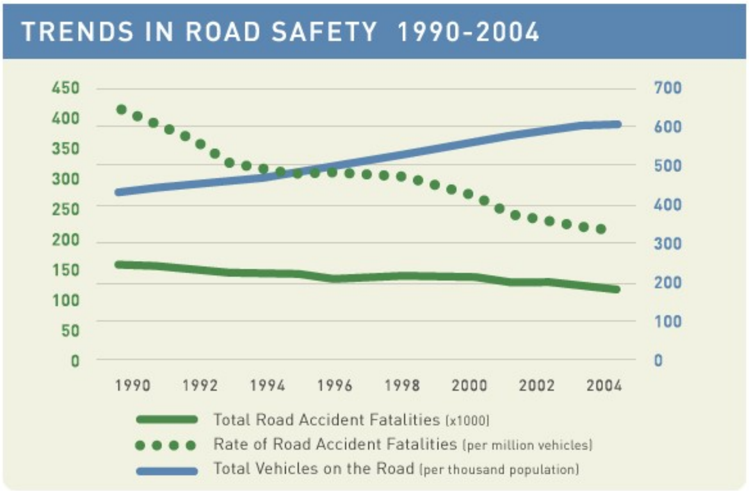


Figure 1.2: The trend on road safety [50]

tional safety. Product safety tackles hazards such as fire hazard, electrical hazards, etc. Functional safety concerns the system behavior in case of failures or faults in the system. Here, a function captures the implementation independent specification of the behavior of the vehicle and its systems. In this thesis, we focus only on functional safety. As by moving towards higher automation levels [111], the impact of functional safety on overall road safety grows further[1]. Achieving functional safety would be the determining factor for the success of Automated Driving (AD) to decrease road fatalities.

The traditional concern of functional safety (covered in ISO 26262) is to design-in safety measures in the system such that after a failure the system is safe or provides a degraded functionality. With the advent of automated driving, a new aspect of functional safety is under development in ISO/PAS 21448 [55]: Safety of the Intended Functionality (SOTIF). In this new aspect, the community tackles the challenge of defining safe functionalities given the limitations in technology and uncertainties in the environment and driving scenarios assuming that there are no failure in the system.

Typically, the development of an automotive system is distributed among many organizations in the value chain of the automotive industry. The distributed development adds another layer to the complexity, as the responsibilities for achieving functional safety also need to be distributed along the value chain. The introduction of smart mobility disrupts the traditional structure of this industry with the OEMs on the top, and Tier companies as suppliers. For example, the role of vehicle to other systems (V2X) communication technologies for achieving connected vehicles requires ICT companies to enter the automotive market. Assurance of achieving predictable functional safety requires a reevaluation of the best practices to better use integrated design approaches; especially because automotive is mostly a self-certified industry, unlike other industries such as avionics or health care.

Companies or other organizations that are involved in the automotive value chain contribute to the development of standards and norms to capture the best practices. The

[1] In this thesis we use the five levels of automation introduced in [111]. We consider Levels 3, 4, and 5 to be higher levels of automation.

ISO 26262 standard [54] captures the state of the art for functional safety for design and analysis in the automotive domain. This standard imposes requirements on the process and system design to ensure functional safety. For instance, it has strict requirements for requirements traceability, which takes substantial effort to achieve. Also, the ISO/PAS 21448 [55] is being developed to capture the safety requirements for higher levels of automation. This standard provides guidelines for an iterative development process where the vehicle behavior is refined to adapt to limitations from the technologies and uncertainties in the environment.

Compliance and adherence to these norms can be seen as another layer of complexity on two accounts: First, the number of norms that companies need to follow increases with new standards such as ISO/SAE CD 21434 [56] on automotive cybersecurity and ISO/PAS 21448 [55] on SOTIF. Second, understanding and applying these norms is demanding. They give requirements for various phases in the life cycle from the conceptual phase until production and deployment; therefore, following these norms have impacts on levels of the system and the organization.

Achieving functional safety is the determining factor for the social acceptance of automated driving and smart mobility technologies. The industry faces many challenges in this regard, from design to assessment and certification of such systems. Safety assurance through standard compliance entails considerations on the system from an early phase in a life cycle of a system. In this thesis, we focus on the conceptual design phase and address some of the challenges for integrating functional safety in the architecture of a safety-related system.

1.2 The Challenges for Achieving Safety

We discussed the automotive trends on increasing demands on functionality and the system complexity from three angles: the number of system elements and underlying technologies, the organizational aspect, and compliance to norms. These trends extend the impact of functional safety on the overall road safety and therefore impose some challenges. Here, we briefly discuss these challenges in relation to functional safety and system architecture.

The ISO/IEC/IEEE 42010 [57] standard defines the architecture description as a work product that expresses the architecture of a system. It includes one or more architecture views that address the stakeholders concerns. An architecture view is governed by a viewpoint that establishes the conventions for constructing, interpreting, and analyzing the view to address concerns framed by that viewpoint. A conceptual model of the architecture description is shown in Figure 1.4.

Given the above definition, we can consider functional safety as a viewpoint that frames functional safety as the primary concern. The ISO 26262 standard [54] specifies the conventions of this view point for the automotive industry and gives guidelines and analysis methods for safety. Given these presumptions, we discuss five challenges for achieving safety in automotive safety-related systems. Figure 1.3 gives an overview of these safety challenges.

First challenge is regarding the process aspect of safety. The development process can be seen as the expected *behavior* of an organization (if we look as the organization as a system with a certain behavior). Predictable achievement of quality requires well-defined development processes. The ISO 26262 standard also specifies many requirements that correspond to the process aspect. For instance, it defines the safety lifecycle, which

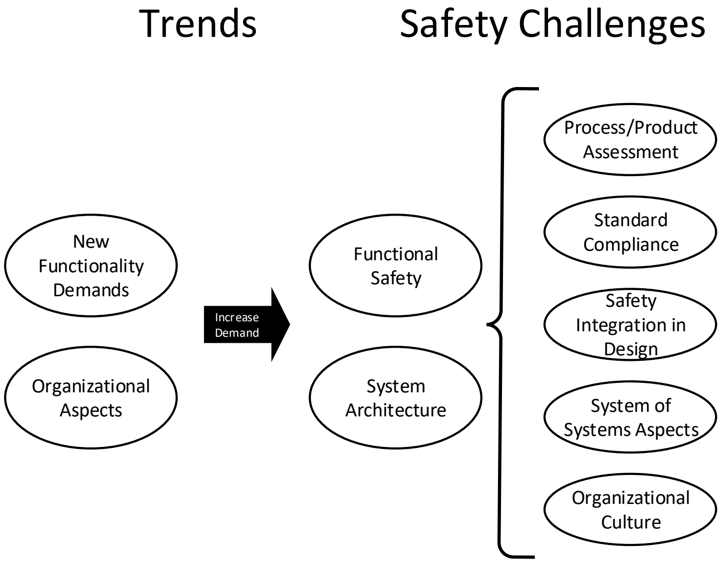


Figure 1.3: The automotive trends and safety challenges

addresses the system from the concept phase to production and service phase. Safety assurance requires assessment of the process side as well as the system design from the technical perspective. Integrating the right processes for developing a complex product can be a challenge. For this integration we need an understanding of the relation between each architecture model with the relevant process steps.

Second challenge is compliance to ISO 26262, which is subject to interpretation and as such its implementation and assessment can be subjective. Given that the automotive industry is self certified in many countries, subjective compliance assessment is a big challenge. As the impact and criticality of the functional safety of in-vehicle embedded systems in achieving road safety increases, objective compliance assessment that does not depend on company policy and culture are crucial. Therefore, novel methods are needed for an objective compliance assessment.

Third challenge is integrating the requirements derived from a safety analysis as well as safety requirements specified in ISO 26262 in the system design (from the design perspective as opposed to development perspective as raised in the first challenge). The ISO 26262 standard is specified to be system independent, it gives generic guidelines such as:

> **Part 4−6.4.4.4** To reduce the likelihood of systematic failures, well-trusted systems design principles should be applied where applicable.

This statement, as vague as it is, avoids giving guidelines on what these well-trusted principles are; and they remain a challenge for complex systems such as automated driving applications. New architectures and designs principles are required for the in-vehicle systems to address these emerging requirements. Achieving these requirements requires interpretation of the norm in the context of the system under development.

Fourth challenge is the considerations of system of systems. Connectivity plays a crucial role in enabling automated vehicles to navigate, as well as in regulating this newly

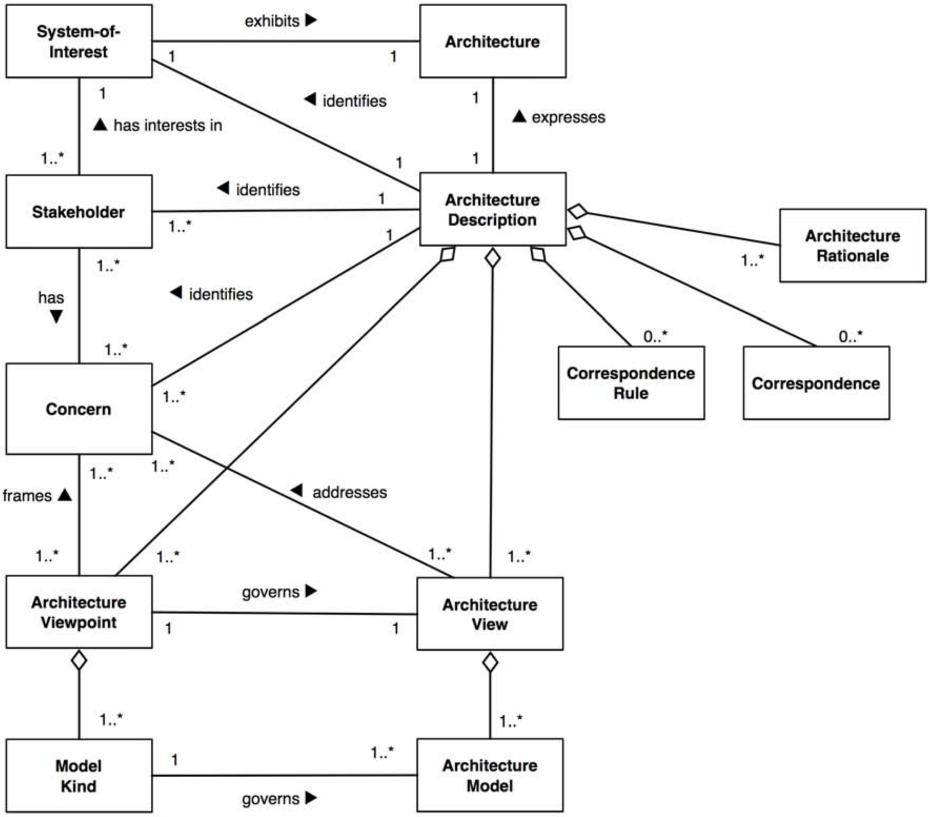


Figure 1.4: The conceptual model of architecture description [57]

established network of connected vehicles as efficiently and safely as possible. As a result, modern vehicles are equipped with vehicle to vehicle (V2V) and vehicle to other systems (V2X) communication capabilities. Vehicles, traditionally considered as a *monolithic* system, now become part of an ecosystem of vehicles, infrastructure and mobility services that can be characterized as a system of systems. We need new safety methods that are applicable to a system of systems.

Finally, we reach to the organizational challenges. People and organizations collaborate to build systems. Their commitment determines success and failure of these systems. As such, we need to consider the human side of design as well. Regardless of development processes and design principles, we need to ensure that the right people with the right mindset are on the job to achieve the right system. Building and maintaining the correct organizational culture that ensures predictable functional safety is a challenge for this industry.

1.3 Research Questions

We formulate some research questions contributing to the functional safety and system architecture challenges discussed above. The primary research objective of this thesis is:

RQ. *How to effectively and consistently integrate functional safety into system development and design in the automotive domain?*

We decompose this primary objective into five more refined research questions. Functional safety assurance and compliance can be subjective and prone to human error due to the inherent internal inconsistencies and possibly vague requirements. Conceptual modeling has been suggested as a solution to overcome these inconsistencies. If we view functional safety as an architectural viewpoint as defined in ISO/IEC/IEEE 42010 [57], we can model the artifacts required for functional safety. In fact, recent research formalized this domain through model-based engineering [4], [7], [12], [76], [85], [124]. However, most research is focused on the system design aspect of safety, and does not offer a solution for integrating the process aspect in the models. Adherence to a defined development process is crucial for achieving predictable quality (in this case, functional safety). We propose **RQ 1** to integrate process and system design systematically.

> **RQ 1:** *How can domain models of functional safety cover both system design and process aspects?*

Safety assurance in the automotive industry is mostly achieved by compliance with norms and standards. Norm compliance system design requires much effort in a thorough systematic analysis of all system parts. Given a formal description of the standard and the models of project artifacts, compliance to the standard can be checked automatically. To study this possibility, we define **RQ 2** as follows:

> **RQ 2:** *How can model-based techniques be used for compliance assurance?*

As system complexity grows, safety assurance becomes dependent on considering the safety concerns in the early development phases. The architectural design of systems determines their quality attributes as well as their capability for addressing non-functional requirements. Architectural patterns provide a fundamental way for classifying designs [57]. We propose **RQ 3** to study architectural patterns for safety.

> **RQ 3:** *How can architectural patterns be used for achieving functional safety in automated driving applications?*

As discussed in Section 1.1, connected and autonomous vehicles are part of the future of automated driving. The V2X communication technologies add a layer in the traffic and transport system that needs to be viewed from the functional safety viewpoint as well. Current functional safety methods consider only the vehicle level as the highest abstraction and focus on resolving the hazards at that level. We need methods that can manage the system of systems aspects of safety. We study this matter by proposing **RQ 4** as follows:

> **RQ 4:** *What is the impact of system of systems composition on safety analysis?*

Finally, we address the organizational aspects of safety assurance in **RQ 5**. The management of functional safety, and ensuring predictable functional safety, depends on the organizational culture and priorities. Since the responsibility for achieving road safety is shifted towards the system, we need to ensure responsible development of said systems. If we want to improve on safety culture we need to measure it. Therefore, we define the following question for measuring safety culture in the organizations that are responsible for system development.

> **RQ 5:** *How to measure the safety culture in advanced development or research organizations?*

1.4 Outline of Chapters

In this section, we outline the remainder of this thesis. We revised each publication for this thesis to reflect our growing insight.

Chapter 2: Safety Driven Development and ISO 26262 In this chapter, we give some background information regarding functional safety. We discuss some of the most important aspects of functional safety from ISO 26262 perspective; namely safety management, development process, architecture design, and safety assurance are presented. This chapter is based on:

> [80] Y. Luo, A. Khabbaz Saberi, and M. G. J. van den Brand. "Safety-Driven Development and ISO 26262," *in Automotive Systems and Software Engineering,* Springer International Publishing, 2019

Chapter 3: A Holistic Safety Domain Model In this chapter, we address **RQ 1**. We propose a holistic domain model that can be used for model-based safety engineering. We systematically analyze the standard text and model the concepts. We model both the system design concepts in the safety domain and the required activities concerning those concepts. Our proposed domain model supports both the system design and process aspects of safety.

Chapter 4: A Model-Based Approach for Compliance Assurance In this chapter, we tackle **RQ 2**. We present an approach that supports the development of standard-compliant systems based on model-based techniques. We use a domain model of ISO 26262 that covers both process and system design aspects on an object level. We then define constraints that define non-compliance to this standard. To prove the concept of our approach, we developed a software tool that automatically checks the constraints before or after the related safety activity. This work is based on:

> [67] A. Khabbaz Saberi, D. van den Brand, and M. G. J. van den Brand "Towards compliance assurance for automotive safety-related development: a model-based approach," *in the Poster Session of the 6th International Symposium on Model-Based Safety and Assessment (IMBSA 2019),* 2019

Chapter 5: Architecture Patterns for Safety We address **RQ 3** in this chapter; we present a novel architecture pattern for safety-related automated driving functions. Additionally, we propose a generic approach to compare our pattern with existing ones. The comparison results can be used as a basis for project specific architectural decisions. This chapter is based on:

[79] Y. Luo, A. Khabbaz Saberi, T. Bijlsma, J. J. Lukkien and M. G. J. van den Brand, "An architecture pattern for safety critical automated driving applications: design and analysis," *11th Annual IEEE International Systems Conference (SysCon 2017)*, 24-27 April 2017, Montreal, Quebec, Canada. p. 261-267

Chapter 6: Design Decisions and Quality Attributes In this chapter, we continue to address **RQ 3**. We share our experience with applying architectural patterns to automated driving systems. We particularly discuss the impact of design decisions regarding the operational design domain on (functional) safety. We provide two automated driving systems as discussion cases and investigate the impact of the operational situation on the safety requirements such as safe state and degraded operating mode. This chapter is based on:

[68] A. Khabbaz Saberi, J. Vissers, F. P. A. Benders, "On the Impact of Early Design Decisions on Quality Attributes of Automated Driving Systems," *8 Apr 2017, 13th Annual IEEE International Systems Conference (SysCon 2019)*, April 2019, Orlando, Florida, USA.

Chapter 7: A System of Systems Approach In this chapter, we take a step for answering **RQ 4**. We investigate the impact of applying safety analysis to an SoS with a conventional, "vehicle-centric" development process. We propose a tailored safety lifecycle based on guidelines of ISO 26262 that is augmented to encompass additional considerations pertinent to an SoS. We performed a comparative study by applying our proposed method as well as the traditional (vehicle-centric) approach as per ISO 26262 for safety engineering of a truck platooning application. This chapter is based on:

[64] A. Khabbaz Saberi, E. Barbier, F. Benders and M. G. J. van den Brand, "On functional safety methods: A system of systems approach," *12th Annual IEEE International Systems Conference (SysCon 2018)* April 2017, Montreal, Quebec, Canada, p. 261-267

Chapter 8: Safety Culture for Research Organizations In this chapter, we introduce a method for measuring the safety culture per ISO 26262. We quantify the safety culture based on participants' response to a questionnaire. We measure several contributing factors such as management commitment, awareness, the flow of information, knowledge, and skills. We performed the survey at the Department of Integrated Vehicle Safety (IVS) of TNO, as an R&D organization, and discuss the results of the survey. This chapter is based on:

[62] A. Khabbaz Saberi, F. Benders, R. Koch, J. J. Lukkien, and M. G. J. van den Brand, "A method for quantitative measurement of safety culture based on ISO 26262," *Evolution of System Safety: Proceedings of the Twenty-Sixth safety-related Systems Symposium, 6-8 February 2018*, York, United Kingdom, p. 203-218, 2018

Chapter 9: Conclusions In this final chapter, we conclude our findings during this research and reflect on the research question.

1.5 The Context of this Research

The entire work that led to the creation of this thesis is done at the Technisch Natuurwetenschappelijk Onderzoeksinstituut (TNO). TNO is an independent research organization that focuses on innovative technologies to positively impact society. One of the major research

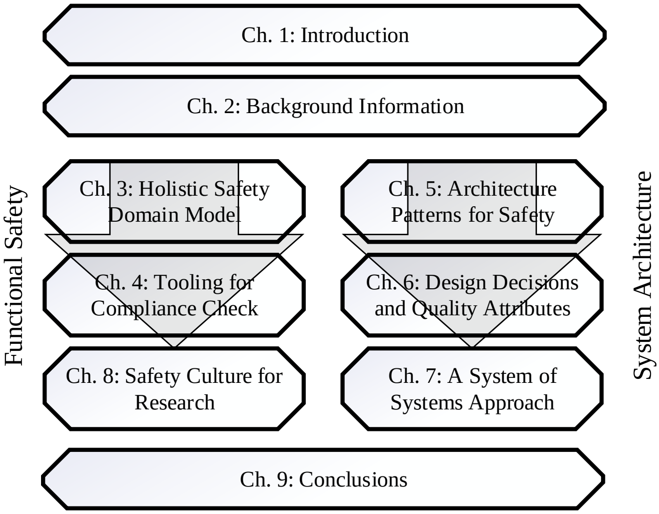

Ch. 1: Introduction
Ch. 2: Background Information
Functional Safety
Ch. 3: Holistic Safety Domain Model
Ch. 5: Architecture Patterns for Safety
System Architecture
Ch. 4: Tooling for Compliance Check
Ch. 6: Design Decisions and Quality Attributes
Ch. 8: Safety Culture for Research
Ch. 7: A System of Systems Approach
Ch. 9: Conclusions

Figure 1.5: The thesis outline

themes at TNO is traffic and transport. This research theme focuses on innovative mobility solutions in order to satisfy the demands of transport and mobility due to increasing urbanization trends.

This research was performed at the Integrated Vehicle Safety (IVS) department of TNO. IVS is actively pursuing innovative technologies, methodologies, and tools for cooperative and automated driving. While the automated driving program focuses on technologies inside the car for automating the vehicle, cooperative mobility focuses on communication as an enabler for autonomous driving. The advantage of cooperative mobility is that automated vehicles can be aware of the circumstances in the environment before arriving in the situation. Therefore, smarter decisions can be made to optimize the traffic flow or avoid any possible hazard.

At IVS, we have many projects that involve the development of safety-related highly automated applications for various vehicle types. We work on systems with a wide range of Technology Readiness Level (TRL) from proof of concepts with very low TRL (in range of 2-3) to implementation ready systems with TRL in range of 6-7. The main motivation of this research comes from these projects that needed a more effective way of considering safety in the design of automated driving applications. In the past, safety was considered secondary to function development, and typically, after developing a particular functionality, we ensured safety by adding additional features. For instance, we would add a collision avoidance feature that triggers emergency braking if a collision is imminent. However, by moving towards demonstrations of automated driving features on the public roads, we were required by RDW [106] (the Dutch national road safety agency) for better assurance of the safety of our systems. There was also an interest to study the application of ISO 26262 for automated driving. Therefore, we felt the need for a better way of considering functional safety requirements in the development of our automated driving application. As a result, this research was initiated to study these new and emerging requirement for safety driven design.

One of the contributing programs to this research was the EcoTwin program. EcoTwin was a four-year program for developing truck platooning concepts [125]. The EcoTwin projects aimed to automate a truck such that it can follow a leader truck through traffic on Dutch public roads. This project was organized in two parts: Technical realization, and Safety. Technical realization part, as the name suggests, was responsible for the design and implementation of the automated driving functions. And, Safety part was responsible for the safety of the design and implementation as well as the legal aspects of the project. There were several partners involved in EcoTwin. In addition to TNO, DAF [17], Ricardo GmbH [107], I&M [90] (the Dutch ministry of infrastructure and the environment), and RDW [106] also had an important role concerning this project. I&M was active in the regulatory aspects of EcoTwin. The ministry was also interested in letting The Netherlands be the first country to have public road testing of automated vehicles. One of the most important objectives of this project was to test the automated driving functions on public roads. To realize this goal, safety played an important role. Ensuring the safety of a complicated system such as the automated truck requires a systematic approach. The ISO 26262 standard was chosen as the main guideline for the development process in this project. With EcoTwins I and II, TNO successfully tested the truck platooning concept trucks on a closed public road. With EcoTwin III, we achieved automation Level 3 (according to SAE standard [112]) and demonstrated the platooning concept on open public road during the European Truck Platooning Challenge [48].

1.6 How to Read This Thesis

This thesis is written as a collection of papers, and each chapter is based on a single publication. As such, chapters are standalone and can be studied separately. The underlying storyline is depicted in Figure 1.5. We start with giving some background information in Chapter 2. Chapters 3, and 4 address the modeling of ISO 26262 and compliance to it. We address the architectural challenges and the application of design patterns in practice in Chapters 5 and 6 respectively. We discuss some of the challenges related to the system of systems aspect in Chapter 7. Chapter 8 addresses the safety culture assessment. Finally, we conclude this thesis in Chapter 9.

In addition to the publications that are reflected in this thesis, our research resulted in few other publications that are not reflected in this thesis. We invite you to read them:

[33] E. de Gelder, J. P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, J. Ploeg, L. Friedman and B. De Schutter. “Ontology for scenarios for the assessment of automated vehicles,” (Submitted to Transportation Research Part C: Emerging Technologies)

[65] A. Khabbaz Saberi, J. Hegge, T. Fruehling, J.F. Groote. “Beyond SOTIF: Black Swans and Formal Methods,” *14th Annual IEEE International Systems Conference (SysCon 2020)* (Accepted)

[61] A. Khabbaz Saberi, A. Smulders, J. J Lukkien. “Towards a Holistic Assurance Methodology: From Component to Information Assurance,” *The Fast Abstract Track of the 38th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2019)*, Finland, 2019

[19] E. de Gelder, A. Khabbaz Saberi, H. Elrofai. “A method for scenario risk quantification for automated driving systems,” *The 26th International Technical Conference and exhibition on the Enhanced Safety of Vehicles (ESV)*, Eindhoven, The Netherlands, June 2019

[84] Y. Luo, M. G. J. van den Brand, and A. Khabbaz Saberi. “A systematic approach and tool support for GSN-based safety case assessment,” *Journal of Systems Architecture: Embedded Software Design : the EUROMICRO*, p. 1-16, May 2017

[128] E. van Nunen, F. Esposto, A. Khabbaz Saberi, and J. P. Paardekooper. “Evaluation of safety indicators for truck platooning,” *2017 IEEE Intelligent Vehicles Symposium (IV)*,Los Angeles, California, p. 1013-1018, June 2017

Chapter 2

Safety Driven Development and ISO 26262

The automotive industry has seen a rapid change in the technologies used inside the vehicles. Since the introduction of the first electronic control unit, the impact of electronics and computer science on the quality of the vehicles are increasing every year. Arguably, safety is one of the most important quality attributes of a vehicle that needs special attention during all the stages of the life cycle of a vehicle. The overall safety of a vehicle has multiple aspects, such as passive safety, active safety, and functional safety. Functional safety addresses the hazards that are caused by the malfunctioning of Electrical and Electronic (E/E) systems. Many factors impact functional safety such as the organization and management, the development process, the design of the systems, the system type and technologies used in it, the quality control methods, etc. The ISO 26262 standard provides the state of the art of functional safety in the automotive industry. In this chapter some of the most important aspects of functional safety from the ISO 26262 perspective are discussed; namely safety management, development process, architecture design, and safety assurance are presented here.

This chapter is based on:

[80] Y. Luo, A. Khabbaz Saberi, and M. G. J. van den Brand. "Safety-Driven Development and ISO 26262," *in Automotive Systems and Software Engineering,* Springer International Publishing, 2019

2.1 Introduction

In safety-related domains such as automotive, railway, and avionics, even a small failure of a system might cause injury to or death of people. A number of international safety standards are introduced as guidelines for system suppliers to keep the risk of systems at an acceptable level [2], such as IEC 61508 (multiple domains) [47], ISO 26262 [52] (automotive domain), DO 178C (avionic domain) [119], CENELEC railway standards (railway domain) [26]–[28].

In the automotive domain, currently, the ISO 26262 standard, which is a goal-oriented standard for safety-related systems within the scope of road vehicles, is state of the art. Since its introduction in 2011, ISO 26262 has attracted more and more attention in the automotive domain. Many safety-driven development methods are proposed based in this standard [66]. In this Chapter we first introduce some basic concepts in the ISO 26262 standard Section 2.1), then we discuss safety management in Section 2.2 and the safety lifecycle (Section 2.3) in the context of the standard. Furthermore, a brief comparison of several safety architecture patterns is given in Section 2.4. Finally, as compliance with safety standards is a basis of safety assessment, some model-driven techniques, designed for supporting safety assessment, are presented in Section 2.5.

2.1.1 ISO 26262

The ISO 26262 standard is an adaptation of the generic IEC 61508 standard, which focuses on Electrical/Electronic (E/E) systems but provides a general design framework for safety-related systems [72]. Similar to IEC 61508, ISO 26262 is a risk-based safety standard. It provides a risk-driven safety lifecycle for developing safety-related systems in the automotive domain. In the standard, the risk of hazardous situations is qualitatively assessed. This assessment is done to avoid or control the systematic failures and to detect or control random hardware failures.

The ISO 26262 consists of ten parts as shown in Figure 2.1. Part 3 to Part 7 correspond to the safety lifecycle, while Parts 1, 2, and Part 8 to Part 10 provide the additional information related to the interpretation of the main parts. The ISO 26262 standard is structured based upon the V-model. Parts 3 to 7 construct the primary V cycle for the whole system development, The main goals of Part 3 is to identify system hazards and risks through Hazard Analysis and Risk Assessment (HARA), define safety goals, and the Functional Safety Concept (FSC). Part 4 focuses on the system level development, integration, and validation. In this part, Technical Safety Requirements (TSRs) are derived based on the FSC.

Moreover, Parts 5 and 6 have their own (smaller) V cycles for hardware and software development respectively. In these two parts, more detailed safety requirements are derived from TSRs. These safety requirements are assigned to concrete subsystems or components for implementation. Finally, Part 7 covers the release of the system for production.

2.1.2 Functional Safety Definition

Functional safety is easy to understand, yet difficult to formally define. The ISO 26262 standard defines functional safety using some other concepts which have complex definitions. An overview of the full definition is shown in Figure 2.2. The definition of functional safety reads as follows:

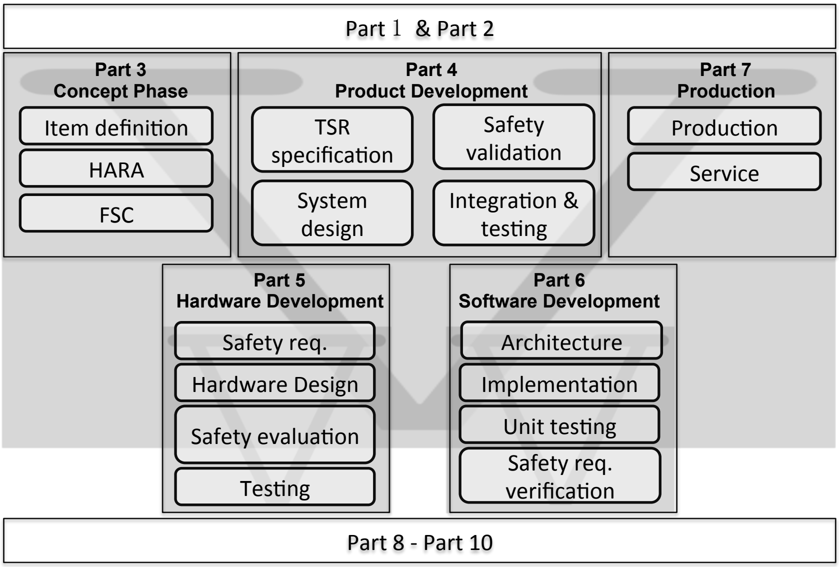


Figure 2.1: An overview of the ISO 26262 V-model [76].

Definition 1. *"Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems."*

The phrases in *italic* are elaborated with more definitions. After compiling all the defined concepts, the definition transforms to:

> "Absence of combination of the probability of occurrence of physical injury or damage to the health of people and the severity of that *harm*, judged to be unacceptable in a certain context according to valid societal moral concepts due to potential sources of *harm* caused by termination of the ability of an *element* or an *item* to perform a function as **required** or unintended behavior of the *item* with respect to the **design intent** for this *item* of systems that consist of electrical and/or electronic *elements*, including programmable electric *elements*."

We can agree that this definition is difficult to grasp at first glance! To add to the complexity, there are also some side notes attached to this definition. For instance, in the definition of *failure* there is an important note about the difference between required, and specified failures. The ISO 26262 standard considers incorrect specification also a source of failure, which is a quite strong definition.

We suggest a simplified yet less accurate approach for defining functional safety to simplify the definition and ease the burden of understanding functional safety. There are a few vital implications in ISO 26262 definition of functional safety, namely:

1. functional safety depends on the design intent,
2. functional safety tackles (in the scope of ISO 26262) failures of E/E systems, and
3. functional safety is only applicable to hazards that cause harm to people (and not damage to property).

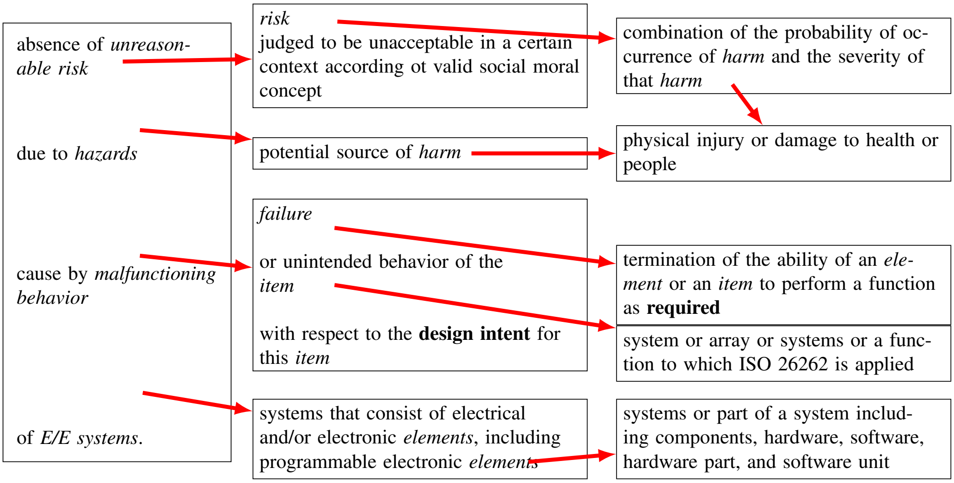

absence of *unreasonable risk*
due to *hazards*
cause by *malfunctioning behavior*
of *E/E systems*.
risk
judged to be unacceptable in a certain context according ot valid social moral concept
potential source of *harm*
failure
or unintended behavior of the *item*
with respect to the **design intent** for this *item*
systems that consist of electrical and/or electronic *elements*, including programmable electronic *elements*
combination of the probability of occurrence of *harm* and the severity of that *harm*
physical injury or damage to health or people
termination of the ability of an *element* or an *item* to perform a function as **required**
system or array or systems or a function to which ISO 26262 is applied
systems or part of a system including components, hardware, software, hardware part, and software unit

Figure 2.2: The overview of functional safety definition in ISO 26262

Considering these points, we propose a shorter definition of functional safety:

Definition 2. *Operating as intended with fail-safe or fail-operational strategies to prevent hazards.*

In this definition, "operating correctly" means doing what is intended, referring to the design intent. In other words, correct behavior reflects the design intent. Moreover, the design should be able to cope with possible failures, thus preventing hazards from happening. In this definition, we identify two major strategies for coping with failures: fail-safe that means reverting to a safe state which no longer provides the required functionality, and fail-operational that means transitioning to a safe state while some variation (which may be a degraded operating mode) of the functionality is still provided.

The proposed short definition, in comparison with the longer definition in ISO 26262, lacks the notion of *risk*. Therefore, this definition does not convey the goal of reducing the risk that is addressed in the long version. However, this definition makes it easier to understand functional safety in a pragmatic manner.

2.1.3 Functional Safety Goals

Failures of the E/E systems are recognized (the focus of functional safety) as the primary cause for hazards. There are various ways for categorization of failures. One generic way is to classify them into two types: random (hardware) failures, and systematic failures. Random hardware failures are unpredictable failures that occur during the lifetime of a hardware part [52]. These failures are only relevant to hardware parts and do not apply to software units. Systematic failures are, on the other hand, deterministic and have a certain cause (usually the design of the system). These failures can happen in both hardware, and software elements. An example of these failures is a software bug/error. An overview of this categorization and the relation between failures and hazards are shown in Figure 2.3.

From the definition of functional safety, it can be inferred that failures (may) cause hazards. Consequently, the goal of functional safety is to reduce the risks of hazards. This

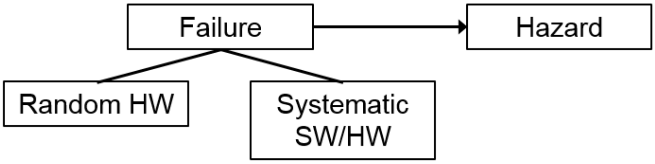


Figure 2.3: The relation between failures and hazard

goal can be refined into two subgoals by categorizing the types of failures: 1. preventing systematic failures, and 2. mitigating random failures.

Preventing systematic failures implies that the development process of safety-related systems should be carried out in such a way that the human errors (i.e., the primary cause of systematic failures) or other contributing factors do not lead to an unresolved failure. This goal is achieved by defining a predictable process for the development of safety-related systems. The mechanism for ensuring the achievement of this goal includes reviewing work-products, analysis, and testing.

We mitigate random hardware failures during design by analyzing possible failures and using detection and reaction mechanisms known as *safety mechanism*. The random hardware failures are a probabilistic phenomenon, and they are unpreventable. Hence, there should be mechanisms in the design that detect these failures and act to prevent failures from creating hazards.

Furthermore, it should be possible to provide evidence about the achievement of the subgoals mentioned above systematically.

In summary, there are three main subgoals for functional safety:

1. preventing systematic failures,
2. mitigating random failures, and
3. showing (providing evidence) that the previous goals have been achieved.

These goals are achieved by a combination of controlling the development process, design, verification and validation, and documentation.

2.2 Safety Management

Safety engineering is complex, especially in a multidisciplinary domain such as the automotive industry. It involves a wide variety of tasks that are typically carried out by multiple people with various skills and experiences. The safety related tasks are the activities that are performed during the safety lifecycle (referred to as safety activities in ISO 26262 vocabulary). As defined by ISO 26262, the safety lifecycle is the entire duration of time that a safety-related system exists, from the concept phase, to the decommissioning phase. Moreover, the safety activities are highly dependent on each other, as well as other non-safety activities related to development, testing, and production of a safety-related system. Therefore, safety management is a necessity to ensure systematic and smooth realization of all the safety related activities. Since management attention is required for the realization of safety activities, ISO 26262 also provides some guidelines on the most critical considerations for safety management.

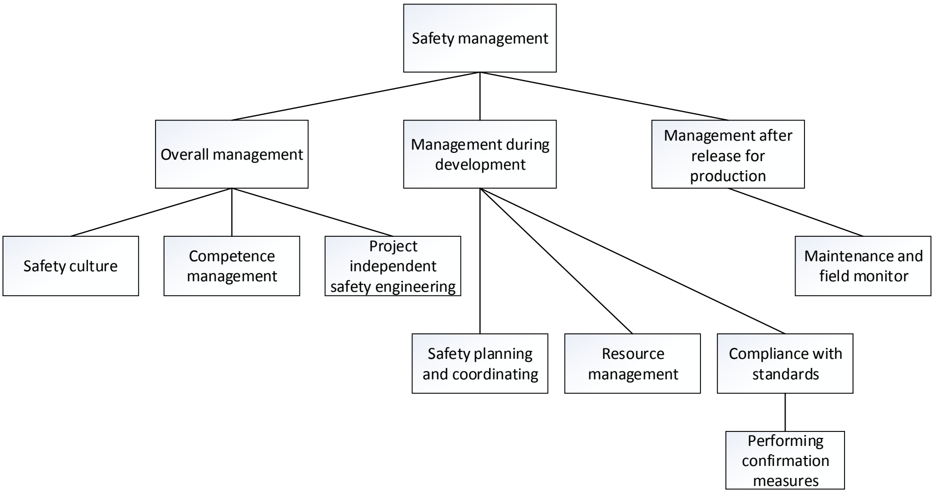

Safety management
Overall management
Management during development
Management after release for production
Safety culture
Competence management
Project independent safety engineering
Maintenance and field monitor
Safety planning and coordinating
Resource management
Compliance with standards
Performing confirmation measures

Figure 2.4: The overview of safety management parts

Safety management is divided in three main parts in ISO 26262 Part 2: overall safety management, safety management during development, and after release for production. An overview of safety management is shown in Figure 2.4. The overall safety management considers the project independent aspects of safety engineering. This includes safety culture, competence management, quality management, and definition of project independent development process. Safety culture has been the main focus of safety engineering in several industries after the Chernobyl disaster in 1986. Safety culture is described in more details in the following subsection.

The goal of safety management during development and after release for production is ensuring safe realization of a safety-related system. This is (in most cases) done by ensuring compliance with a safety standard such as ISO 26262.

The ISO 26262 standard recommends assignment of a safety manager to the development of a system. The primary goal of the safety manager is to coordinate all safety activities during the safety lifecycle. Planning and coordinating of safety activities, and resource management are typical responsibilities of safety managers.

The responsibilities of a safety manager overlaps with those of a project manager, in tasks such as planning, and resource management. The difference is that the safety manager is involved only with the safety related planning, and resources in the overlapping tasks. There are tasks with no overlap for these two roles too. For example, costs management is only a task of the project manager. Another example is risk management, and project control. While both the project manager and the safety manager perform these tasks, yet they have different focus. The project manager performs risk management for project risks, whereas the safety manager cares for the system safety risks. Similarly, both roles perform project control, but the safety manager cares only about control mechanisms that impact safety. The ISO 26262 refers to these control mechanism as *confirmation measures*. These activities are performed to increase the trust in the development process with respect to safety related issues. Confirmation measures are described in more details in the rest of this chapter.

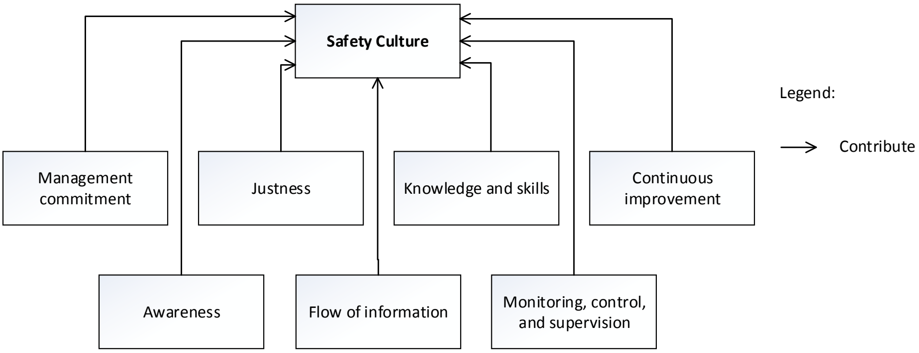


Figure 2.5: Safety culture contributing factors

Lastly, safety management after release for production is responsible for planning of maintenance and field monitoring for possible undiscovered failures.

2.2.1 Safety Culture

Safety culture is one of the key elements of overall safety management in ISO 26262 [52]. In general safety culture requires the organization to provide the proper environment for people involved in safety activities. Safety culture is defined by [101] as follows:

> “The set of enduring values and attitudes regarding safety issues, shared by every member of every level of an organization. Safety Culture refers to the extent to which every individual and every group of the organization is aware of the risks and unknown hazards induced by its activities; is continuously behaving so as to preserve and enhance safety; is willing and able to adapt itself when facing safety issues; is willing to communicate safety issues; and consistently evaluates safety related behavior.”

An overview of the contributing factors to safety culture is shown in Figure 2.5. These key factors are the result of aggregation of the aspects considered in the literature [101], [131]. The description of these factors are as follows:

Management commitment is the willingness of the organization at every level (from top to down) to invest effort in safety and their genuine positive attitude towards safety. The ISO 26262 standard emphasizes on this factor in Part 2: 5.4.2.1 and 5.4.2.2.

Justness (only considered in [101]) is the extent to which behavior according to functional safety is encouraged and rewarded by the organization. Moreover, there should be a “no blame” culture where in event of an accident, solutions are sought instead of blaming the responsible person. The ISO 26262 also mentions this matter in Part 2: 5.4.2.1.

Awareness is the level of individuals’ appreciation of their role and impact on functional safety, and on safety in general. Moreover, the understanding of the risks involved in their work for themselves and others is also a part of awareness. The ISO 26262 standard addresses the issue of roles in Part 2: 5.4.2.2.

Flow of information is the accessibility of new information for the right people through transparent communication. For instance, if there is a new hazardous situation identified during a recent test, the information should be easily provided to others, to be considered

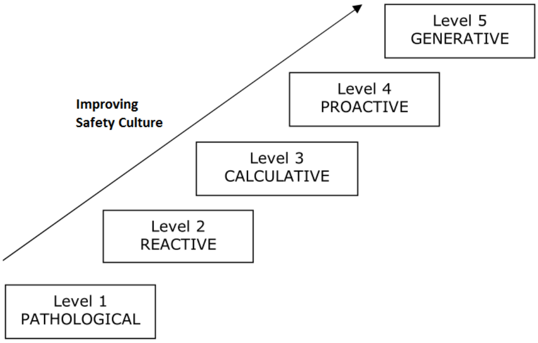


Figure 2.6: Safety Culture Maturity Model [101]

if applicable in their projects. In ISO 26262-2 5.4.2.3 the flow of information is mentioned as explicit communication of functional safety anomalies. The ISO 26262 standard even takes flow of information further by stating that there should be a process for resolving functional safety anomalies in Part 2: 5.4.2.4.

Knowledge and skills (similar to "behavior" in [101]) are the extent of individuals' knowledge of safety engineering processes and activities, and in particular in this case, the ISO 26262 standard. This factor is more important in a research and development environment. General appreciation of the relevant knowledge and skills are needed in an organization to allow effective implementation of functional safety. Several clauses of ISO 26262 can be linked to this aspect of safety culture such as Part 2: 5.4.2.5, and 5.4.2.6.

Continuous improvement (the same as "adaptability" in [101]) is the willingness of an organization to learn from their experiment and improve on the way of working of the organization. Continuous improvement is also mentioned in Part 2: 5.4.2.7.

Monitoring and control (only considered in [131]) is the existence of supervision mechanisms concerned with safety and the visibility of these mechanism in the organization. Moreover, the extent of availability of the required authority to execute functional safety activities is also part of this aspect. The supervision issue can be traced in ISO 26262 Part 2: 5.4.2.8.

2.2.2 Safety Culture Metrics

A model for safety culture maturity is introduced by [46]. Other related work has been done by [30] on safety culture maturity model. An overview of the maturity model is shown in Figure 2.6. Similar to the Capability Maturity Model (CMM), the safety culture maturity model has five levels. The general idea is that an increase in the level shows improved safety culture maturity.

The first level, indicating the worst safety culture, is when an organization considers safety as a burden. There are typically no processes in place for handling safety issues, and the members of the organization only care about not getting in trouble. The second level is applicable when there are some processes for safety but not strictly followed by the members. It could be that the management of the organization states that safety is important, but it is not believed by the members. In the next level, i.e. the calculative level, the safety processes are followed and the members are more involved in the safety issues.

Nevertheless, the safety processes are not believed to be critical. In the proactive level, both the management and the members believe in their safety processes, and all hazards are addressed systematically. In the last level, safety is deemed an organization value. Both members and management are constantly improving the safety. More details can be found in [44].

The ISO 26262 standard does not provide any recommendations on the safety culture maturity level. Therefore, it is the companies' ambition that drives the target with respect to their safety culture maturity level. Identifying the safety culture maturity level of a company, and maintaining or improving it is therefore also the responsibility of that company.

Depending on the level of safety culture maturity, the actions needed for improving or maintaining the safety culture differ. Changes in areas such as management support, processes, training are required for improving the safety culture. More information on this topic can be found in [45].

2.2.3 Confirmation Measures

Ensuring compliance with safety standards is one of the important responsibilities of the safety manager during development. The ISO 26262 standard created mechanisms, referred to as confirmation measures, for ensuring compliance with this standard. These measures include confirmation reviews on a selected work products indicated by ISO 26262, functional safety audit, and functional safety assessment. Depending on the Automotive Safety Integrity Level (ASIL) assigned to the system of interest, these measures shall be performed by the indicated people. This indication can be a different person (than the creator of the work product), a person from a different team within the same organization, or a person from an independent (with respect to management structure) organization. Some examples of the confirmation measures are shown in Table 2.1.

2.3 Safety Lifecycle: Integrated V model

The ISO 26262 standard is the collection of best practices in the automotive industry; thus, it has a number of practical considerations that are specific to this domain. The best example of these considerations is the differentiation between the functional safety concept, and the technical safety concept. These two phases correspond to the functional view and physical view in system engineering development. The reason for separating these two phases, which in other domains are carried out in parallel, is due to the special considerations between the Original Equipment Manufacturers (OEMs), and their suppliers (Tiers). In case of different settings for the development chain, or developing a Safety Element out of Context (SEooC), there is a possibility of using a more effective development process by tailoring the safety lifecycle.

The ISO 26262 standard does not provide recommendations for a development process of the functionality required for the system under development. It defines a safety lifecycle that contains all the activities related to functional safety. Indeed, there is an underlying assumption about another process (seemingly going on in isolation from the safety lifecycle) for designing the system. This process is referred to as Quality Management (QM) in ISO 26262. The safety lifecycle of ISO 26262 requires some information about the functionality of the system from an external process. For instance, the preliminary architecture which is a prerequisite for the functional safety concept needs to be provided via an exter-

Table 2.1: Example of confirmation measures, and ASIL dependent independent level [52]

Confirmation measures	**Degree of independency** applied to ASIL				**Scope**
	A	B	C	D	
Confirmation review of the hazard analysis and risk assessment of the item (see ISO 26262-3:2011, Clauses 5 and 7, and, if applicable, ISO 26262-8:2011, Clause 5) Independence with regard to the developers of the item, project management and the authors of the work product	I3	I3	I3	I3	The scope of this review shall include the correctness of the determined ASILs and quality management (QM) ratings of the identified hazardous events for the item, and a review of the safety goals
Confirmation review of the safety plan (see 6.5.1) Independence with regard to the developers of the item, project management and the authors of the work product	—	I1	I2	I3	Applies to the highest ASIL among the safety goals of the item
Confirmation review of the item integration and testing plan (see ISO 26262-4) Independence with regard to the developers of the item, project management and the authors of the work product	I0	I1	I2	I2	Applies to the highest ASIL among the safety goals of the item

The notations are defined as follows:

—: no requirement and no recommendation for or against regarding this confirmation measure;

I0: the confirmation measure should be performed; however, if the confirmation measure is performed, it shall be performed by different person;

I1: the confirmation measure shall be performed, by a different person;

I2: the confirmation measure shall be performed, by a person from a different team, i.e. not reporting to the same direct superior;

I3: the confirmation measure shall be performed, by a person from a different department or organization, i.e. independent from the department responsible for the considered work product(s) regarding management, resources and release authority.

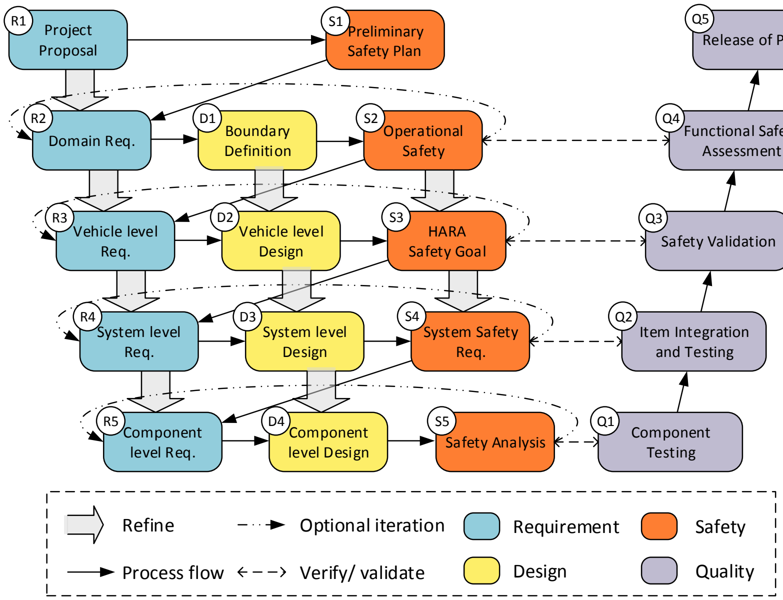


Figure 2.7: The integrated V model

nal source (ISO 26262 Part 3: 8), or (non)functional requirements should be included or referenced in the specification of technical safety requirements (ISO 26262 Part 4: 6.4).

This means that the whole development of a system (both functionality and functional safety) cannot be solely based on the safety lifecycle recommended in ISO 26262. The safety lifecycle addresses the functional safety related developments, yet the functionality of the system is not addressed in this process. Therefore, there needs to be a development process in which the desired functionality is considered.

Considering the mentioned issues, there is a need for alignment of the safety process and the engineering process that creates the functionality of the system.

We propose a model of the integrated V model for functional safety as shown in Figure 2.7. The color codes in the model are used to differentiate between functional, and safety perspectives of development: blue and yellow colors are used for functional design, orange is used for safety parts, and violet is used for verification and validation activities related to both functional and safety parts of the design.

In the integrated V model the requirements development is modeled in a separate flow (as opposed to the traditional V model) to emphasize the hierarchical structure of requirements, and to enforce gradual development and refinement of requirements based on higher level requirements and design.

The safety lifecycle of ISO 26262 is simplified in various ways in the integrated V model. To start with, the production phase of ISO 26262 is removed completely; which in turn, reduces some related activities too. Furthermore, the safety requirements hierarchy is slightly modified by merging functional safety requirements and technical safety requirements into system safety requirements. In addition, the dual V model (Vee of Vee) of ISO 26262 is reduced to a single V model. In other words, the development of hardware, and software (which is followed in separate V models in ISO 26262 as shown in Figure 6) is merged in the main V model; this change results in a reduction of verification and validation activities too.

This process model is specially useful for non-conventional automotive companies who do not require full ISO 26262 compliance; however need to include safety considerations in their development processes. simply since this process does not reflect the norms within automotive industry. Moreover, it could also be useful for conventional automotive companies that require a light-weight process for a specific project. This can be because of different reasons, for example: projects with a tight time to market requirement that does not allow full process coverage, or early development projects that require tight coupling of design and safety processes. The description of the steps of the integrated V-model is as follows:

Project proposal (R1): Our proposed integrated V model starts with the project proposal, in which the general goals of the project, customer wishes, application, the project business plan, etc. are reflected. This step also contains the planning of the development activities for designing the system under development.

Preliminary Safety plan (S1): In this step, a preliminary safety plan is made according to ISO 26262 guidelines. The safety plan contains the planning of all the safety activities related to the safety of the system.

Domain requirements (R2): Here, the domain requirements as well as customer wishes are described (refined) in the form of high level requirements.

Boundary definition (D1): Based on the high level requirements, the design steps are initiated by defining the system. In this step, the system is defined in interaction with its environment.

Operational safety (S2): Following the definition of the system, operational safety starts where the safety critical behavior of the item is defined. It should be noted that the operational safety is not part of ISO 26262 safety lifecycle. The goal of operational safety is to deduce the high level nominal behavior requirements of the item from a safety point of view. At this point iterations over high level steps are made in order to reflect the possible changes that may be needed for satisfying operational safety requirements. After completion of this step, functional safety assessment (Q4) should be planned.

Vehicle level requirements (R3): Here, the vehicle level requirements are defined by *translating* user's wishes and the high level requirement to functional and non-functional vehicle level requirements. Moreover, the requirements based on the boundary of the system are also addressed at this point.

Vehicle level design (D2): During vehicle design, the internal functions are designed to address the vehicle level requirements and operational safety requirements. D2 is the equivalent of functional architecture design (the same as preliminary architecture in ISO 26262).

The combination of the steps R2, R3, D1, D2, and S1 composes the item definition from ISO 26262.

Hazard analysis and risk assessment, and safety goals (S3): This step is performed following the guidelines of the standard and results in safety goals. The resulting safety goals may need iteration over the vehicle level design. When the vehicle level steps are finished, test cases should be designed based on functional requirements and safety goals to be performed for system/safety validation (Q3).

System level requirement (R4): Afterwards, the system level requirements are described, containing both functional and technical requirements, by refining the higher-level requirements.

System level design (D3): Next, the design is further refined by system level design. The architecture designed in D2 is detailed to satisfy the requirements from R4 and S3.

System safety requirements (S4): Following the system level design, system safety requirements are described by refining S3 based on D3. Moreover, the system safety requirements are verified by doing qualitative functional safety analyses such as FMEA, FTA, etc. Similar to previous levels, iterations are made after S4 for revising the design in D3 with respect to requirements in R4.

Item integration and testing (Q2): Afterwards, integration tests should be designed based on the system level design and system level (safety) requirements.

Component level requirements (R5): During this step, the hardware and software requirements refine both the system level safety and non-safety requirements (S4 and R4).

Component level design (D4): In the component level design, the components of the system are detailed and implemented.

Safety analysis (S5): In safety analysis step, the functional safety analyses are performed on the system. The analysis is used to verify the safety of the system in a quantitative manner.

Component testing (Q1): The components tests are designed in this step based on requirements in R5.

Quality (Q1-Q4): Finally, following the steps Q1-Q4 verifies and validates the design against the requirements, and the system is ready for delivery in Q5.

The proposed integrated V model introduces a simplified version of the ISO 26262 standard lifecycle matched with a development process. Synchronization points between the two processes are clearly defined. Furthermore, iteration points within the same level are defined. Additionally, the defined design levels facilitate hierarchical architecture design, and requirements elicitation.

2.4 Safety Architecture Patterns

Besides the development process (discussed in the previous section), which is the major contributor to the first goal of functional safety (preventing systematic failures), the architectural design also has an important role in achieving the first two goals. Good architecture design reduces the chances of making mistakes (primary source of systematic failure) during implementation. Moreover, it can provide proven solutions for mitigation of random failures. Therefore, in this section, this topic is discussed and some guidelines for choosing a suitable architecture pattern are proposed.

Decisions about the system architecture have a great impact on characteristics of the system under development. Furthermore, since architectural design is usually done at the early stages of a project, it is important to consider safety, and specifically functional safety in the design. One of the recommendations of ISO 26262 is to use well-trusted design principles for system architecture. Traditionally, architectural principles are stated using architecture patterns or styles [13]. The ISO/IEC/IEEE 42010 standard [57] recognizes architecture patterns as a fundamental mean for expressing design. Moreover, adherence to architecture patterns is considered as a form of redundancy in other domains too [127].

There are various architecture patterns for safety critical systems in the literature [22], e.g. Protected Single Channel, Homogeneous Redundancy, Heterogeneous Redundancy, Safety Executive, and 3-level Safety Monitoring (also known as E-Gas). An analysis on the impact of these safety pasterns on cost, reliability, safety, negotiability, and execution time has been provided in [1].

The Protected Single Channel Pattern improves safety by monitoring the input data and checking the data integrity and optionally monitoring the outputs. The Homogeneous Redundancy Pattern improves safety and reliability by copying the main channel and switching between them in case of failure. Duplex, Triple Modular, etc. are different variations of this pattern. The Heterogeneous Redundancy is similar to Homogeneous Redundancy except that each added channel is developed independently, therefore it is one of the most expensive patterns. The Safety Executive Pattern can switch to a secondary channel to bring the system to safe state in case of a failure in main channel. The 3-level Safety Monitoring Pattern is widely used in the automotive industry because of it provides a cost-effective safety solution. This pattern monitors the internal states of a system in the first level, and monitors the inputs and outputs in the second level. The third level is dedicated to the nominal functionality of the system.

2.5 Model Driven Design for Safety Assessment

In the previous sections we discussed the main factors for achieving the first two goals of functional safety. In this chapter we explore the third goal. We specifically discuss how to provide evidence for achieving the first two goals.

The safety standards describe generalized approaches to identifying hazards and risks, design lifecycle, and analyses and design techniques. Therefore, when applying such standards for a specific application, a significant degree of interpretation of those standards may be necessary.

The process for developing safety-related systems in these safety domains is manually checked for compliance with the standards. This checking process is referred as safety assurance and certification. Due to the amount of manual work involved, safety assurance is usually costly and time-consuming. Moreover, when a system evolves, some of the existing safety-assurance data needs to be regathered or re-validated. To address this, model-driven techniques have been applied to facilitate safety assurance. We divide these techniques into three categories: modeling safety standards, modeling safety argumentation, and modeling support for safety case assessment.

2.5.1 Modeling Safety Standards

Models of safety standards are widely used for understanding and communicating among engineers and software developers. However, there are a number of significant challenges to deal with. Firstly, the modeling process suffers from subjectiveness issues. In some domains (such as the automotive domain), there is no authority providing an interpretation of the safety standard, and the modeling process is mainly performed by experts based on manufacturer requirements to ensure sufficient quality. Thus, the whole process of extracting information from the safety standards becomes subjective. Furthermore, when a new version of the standard is released, the models need to be updated or modified. Due to the invisible modeling process, most of the previous work needs to be redone. Secondly, standards are represented in natural language, with the resulting inevitable manual work of interpretation becoming more costly and less reliable. It also increases the difficulty of identifying the reusable information from the safety-related artifacts developed during the safety lifecycle. Thirdly, standards themselves contain inconsistencies. There are a number of synonyms used in the standard, which makes it impossible to generate the models from the standards automatically. Sometimes, standards are even in contradiction with themselves [121]. For example, in ISO 26262, formal methods are merely recommended, while the use of semi-formal methods is always highly recommended. However, the standard does mention formal methods and formal notations at a number of places. Finally, any formal model should support the demonstration of compliance with the safety standard, both for the development process and for the diverse artifacts created during product development. We advocate that standards need to be universally understandable and expressed in a language that is simple, well structured, but strict. For this goal, we believe that in the future it should be possible to transform standards into models automatically, and vice versa.

Work to date has generally involved conceptual modeling of standards for understanding. A conceptual model for the aeronautic standard DO 178B is created to improve communication and collaboration among safety engineers and software engineers [140]. A conceptual model of the generic standard IEC 61508 for electrical and electronic equipment is proposed for the development of compliant embedded software [99]. Also, a study on process modeling has been done in the context of ISO 26262 [70]. All of these studies refer to compliance with the standards from a specific point of view. However, the modeling process is still subjective, which may lead to inconsistencies of the models after future modifications. Furthermore, the *traceability* of the source of the models is not covered: no one knows where the concepts and relations in the models come from, except the expert who has identified or defined them.

To address this, three kinds of models are proposed for the safety standards. The structure model and the conceptual model are introduced to support unambiguous understanding of the standard; the process model supports the demonstration of compliance of the process of the project with the process described in the standard. Due to the different characteristics and aims of ISO 26262 models (structure model, conceptual model, and process model), different methods are chosen to extract and describe these models. Most of the selected description methods in Table 2.2 are widely used in the industry.

The structure model of the standard can be obtained by modeling the table of content. For the conceptual model, we defined the Snowball approach for extraction [83]. The results of the structure model and conceptual model can be represented as an Ecore model, an UML model, or an ontology. For the process model, we have used Software & Sys-

Table 2.2: Methods and tools used for each model.

Model	Purpose	Extraction method	Possible description methods
Structure Model	showing the structure of the standard	Manual modeling of the table of content	UML, Ecore, Ontology
Conceptual Model	capturing the main concepts or terms used in the standard and their relations	Snowball approach	UML, Ecore, Ontology
Process Model	demonstrating the required process described in the standard	Mapping between standard concepts and SPEM elements	SPEM, BPMN

tems Process Engineering Meta model (SPEM) [95] as the description language and the SPEM supporting tool Eclipse Process Framework (EPF) [23] for visualization. Besides, other formal process languages can also be used for constructing process model, such as BPMN [134].

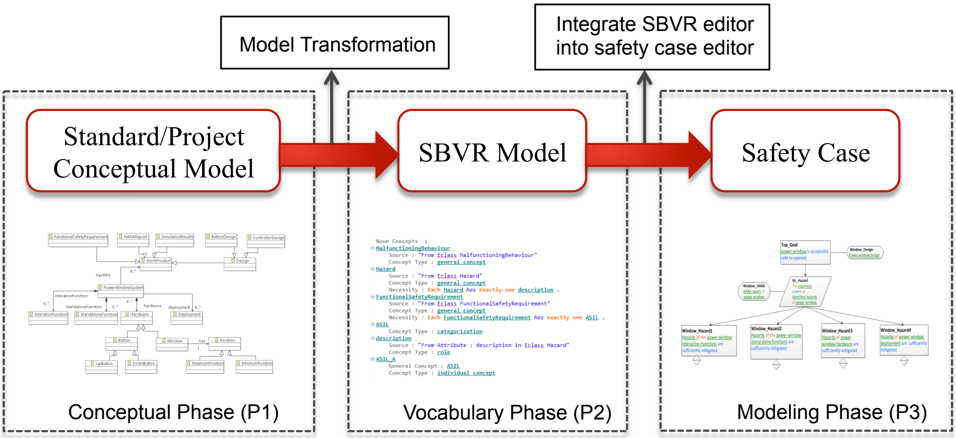
2.5.2 Modeling Safety Argumentation

A safety case is a well-structured argument for justifying that a system is safe. In [10], a safety case is defined as:

Definition 3. *"A documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment".*

In some international safety standards, explicit safety cases are required for safety-related systems. For example, ISO 26262 stimulates the use of safety cases to demonstrate the functional safety [114]. Besides, MOD Def Stan 00-55 [91] for safety-related software in defense equipment requires producing safety cases with explicit safety requirements.

Typically, safety cases are represented in free text, but in this way, the structure of the safety cases might be unclear, which allows for inconsistencies and confusion [81], [82]. To address this, modeling techniques are introduced to facilitate safety case construction and to increase the understandability and confidence in the claimed safety assurance [114]. For instance, techniques originally from model-driven development are used for representing concepts in safety cases, such as ontologies, and Semantic Business Vocabulary and Rules (SBVR) models. Goal Structural Notation (GSN) is introduced as a graphical modeling approach for safety case construction [59]. With the increase of safety-related software and systems, such as cars, more and more GSN-based safety cases are developed. The re-usability of GSN-based safety cases becomes another challenge. People want to reuse safety cases whenever it is possible. Informal reuse of safety case elements occurs, like 'Copy and Paste' of the textual safety case documents between projects. A number of problems with informal reuse are listed in [58]. For example, it may cause inappropriate reuse, lack of traceability, or lack of consistency. To prevent these problems, safety case patterns are introduced as an approach to reuse of common structures of safety cases.



```
Noun Concepts  :
⊖MalfunctioningBehaviour
     Source : "From Eclass MalfunctioningBehaviour"
     Concept Type : general concept
⊖Hazard
     Source : "From Eclass Hazard"
     Concept Type : general concept
     Necessity : Each Hazard has exactly one description .
⊖FunctionalSafetyRequirement
     Source : "From Eclass FunctionalSafetyRequirement"
     Concept Type : general concept
     Necessity : Each FunctionalSafetyRequirement has exactly one ASIL .
⊖ASIL
     Concept Type : categorization
⊖description
     Source : "From Attribute : description in Eclass Hazard"
     Concept Type : role
⊖ASIL_A
     General Concept : ASIL
     Concept Type : individual concept
```

Figure 2.8: An overview of the methodology

2.5.2.1 Safety Case Construction with Controlled Language

As more and more users (argument readers and writers, such as safety engineers, or safety assessors) are involved in safety case development, common understanding of the meaning of a safety case element is important. If it is not the case, the confidence of a safety case can be misplaced. To address this, some research has been done on the understandability of safety arguments. In [40], assured safety arguments are proposed as a clear argument structure to demonstrate how to create clear safety arguments. Besides, in [38], a precise definition of context in GSN arguments is proposed to achieve a better understanding. However, the content of a safety case element is still documented by natural language. The ambiguities caused by using a natural language are still unsolved.

We have proposed a methodology to use an SBVR-based controlled language [94] to support the development of clear safety arguments [77]. By using a controlled language, all the concepts (noun concepts and verb concepts) in a safety case are well-defined in a SBVR vocabulary. Argument readers can check the definitions or examples of those concepts to get a common understanding of them. In this way, the understandability of safety arguments can be improved. Besides, a model transformation has been implemented to generate SBVR vocabularies from EMF conceptual models, which can be obtained via modeling safety standards (Section 2.5.1). Then the manual work involved in vocabulary development can be reduced.

An overview of the proposed approach is shown in Figure 2.8. There are three phases: Conceptual Phase (P1), Vocabulary Phase (P2), and Modeling Phase (P3). In the conceptual phase (P1), a conceptual model of the target domain will be manually built from scratch using the Snowball approach [83] or semi-automatically refined from other conceptual models [78]. The conceptual model will be used as an input of the vocabulary development. The meta model that we use for describing conceptual models is the Ecore meta model. After this, a model transformation will be carried out to transform the conceptual model from an EMF format to an SBVR specification. Then in the vocabulary phase (P2), users (argument writers) can build their own vocabulary based on the generated SBVR model. Note that users can also skip the previous phase and start by creating a new SBVR vocabulary. Finally, in the modeling phase (P3), the vocabulary will be used to

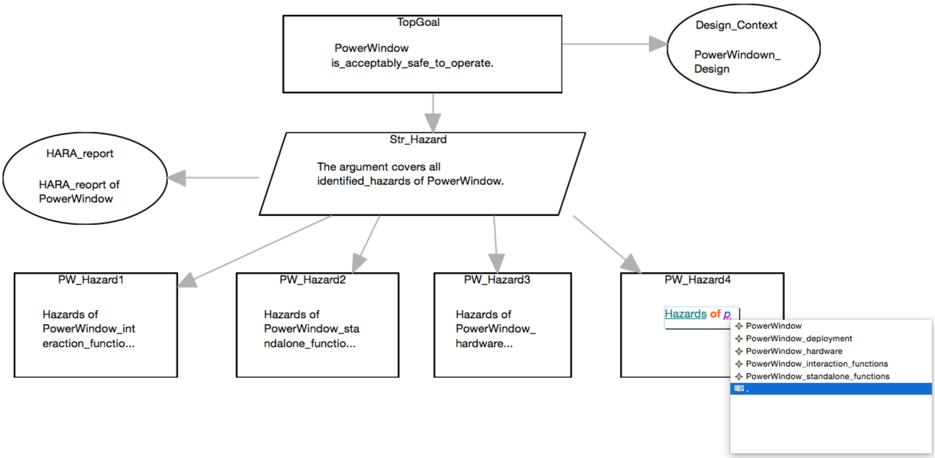

TopGoal
PowerWindow is_acceptably_safe_to_operate.
Design_Context
PowerWindown_ Design
Str_Hazard
The argument covers all identified_hazards of PowerWindow.
HARA_report
HARA_reoprt of PowerWindow
PW_Hazard1
Hazards of PowerWindow_int eraction_functio...
PW_Hazard2
Hazards of PowerWindow_sta ndalone_functio...
PW_Hazard3
Hazards of PowerWindow_ hardware...
PW_Hazard4
Hazards of p
PowerWindow
PowerWindow_deployment
PowerWindow_hardware
PowerWindow_interaction_functions
PowerWindow_standalone_functions
.

Figure 2.9: Illustration of the GSN editor with SBVR functionality

facilitate the safety case construction.

2.5.2.2 A GSN Editor with SBVR Functionality

To construct safety cases in GSN with vocabulary support, we have integrated the SBVR functionality into the GSN editor. As the result, the noun and verb concepts defined in a vocabulary will be highlighted while safety engineers edit a GSN element. Figure 2.9 shows a screenshot of the GSN editor. When a GSN element is edited, a list of suggested concepts is given via content assistant. For example, after typing “p”, a list of concepts in the vocabulary that start with “p” is provided. In this way, the number of errors, such as ambiguities of a safety case can be reduced. Users can always look into the vocabulary to check the definitions of nouns and verbs used in their safety cases to avoid misunderstanding.

2.5.3 Safety Case Assessment

Currently, different industries have different processes for assessing safety cases. To the best of our knowledge, there does not exist a general and formal manual which describes how a safety case is assessed. After a restrict literature study, we have found four sources that have mentioned safety case assessment from different angles. These descriptions are not only specified for GSN-based safety cases, but also applicable for textual safety cases.

2.5.3.1 Overview of Safety Assessment Approaches

The first source is a safety case assessment manual for Gas Conveyors’ Safety Cases provided by HSE [43]. In this document, they described a framework for assessing GSMR (Gas Safety Management Regulations) safety cases. In the Gas & Pipelines Unit, a safety case is assessed in two stages. The first stage is the registration stage. After a safety case has been received by the administration team (AT), the member of the AT checks whether the case is complete as described by its own contents list. Then they initially review the safety case to determine whether it is reasonable and contains sufficient information for

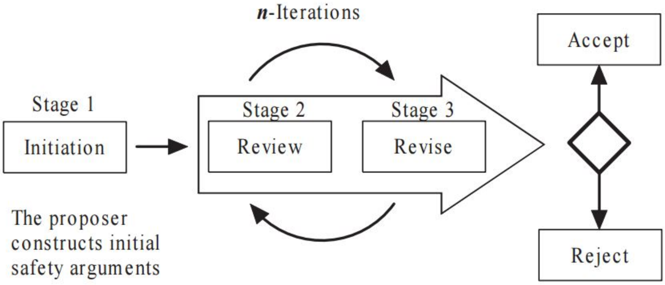


Figure 2.10: Safety Argument Review Process [139]

assessment. The second stage is the main part of safety case assessment. During this stage, the assessor should complete the following steps [43]:

- *Identify, clarify and priorities issues which should be examined further and/or resolved as part of the assessment process;*
- *Discuss and resolve such issues with the proposer of a safety case;*
- *Reach formal agreement on improvements required;*
- *Reach a decision, where possible, to accept the safety case and record why;*
- *Provide reasons, in writing, for rejecting a safety case;*
- *Identify inspection topics.*

The second source is a safety case review process introduced in [138] [139] (Figure 2.10). This process includes three stages: Initiation, Review, and Revise. Stage 1 is the initial development of a safety case which is done by safety case developers. When the safety case is completed, developers submit it to safety case assessors. Assessors need to review the safety case and give feedback for revising it if necessary. This review and revise stage may be repeated several times until a judgment is proposed. The judgment can be either “Accept” or “Reject”.

The third source is the description about the goal and needs of safety assessor in high-level requirements [96] of OPENCOSS. The goal of safety assessors is “*to assess whether a safety demonstration of a product, or assurance demonstration of a system or component is acceptable*”. The principal needs of a safety assessor are “*to view the baseline artifacts of safety case, to improve locating deficiencies and inconsistencies in the safety critical system and to cooperate with safety managers or other safety case assessors*”.

The fourth resource is a presentation by Comentor on the 3rd Scandinavian Conference on System and Safety [98]. It describes three topics around functional safety: the objectives and outcomes of safety case assessment, the benefits of delivering a good safety case and the tips for developing effective argumentation. They stated, “*The objectives of safety case assessment are to evaluate whether the reasoning about the functional safety of the product is valid and to get an independent statement that the claim about the functional safety is reasonable*”. This is consistent with the objectives in the third source. Simply, the assessor is required to evaluate a safety case, then to provide a recommendation which gives judgments on the claims. The outcomes of a safety case assessment could be: identified strengths and weaknesses of a safety case, a recommendation of the judgments, and required corrective actions.

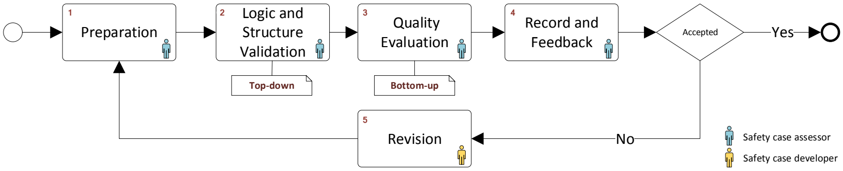


Figure 2.11: The Proposed Safety Case Assessment Process

2.5.3.2 An Alternative Safety Assessment Process

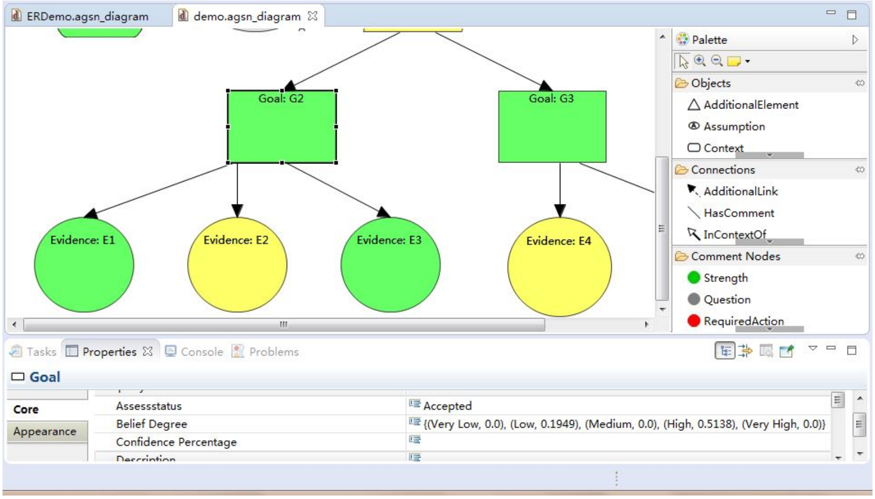
From the aforementioned four resources, we obtained an insight of safety case assessment process. This helped us to understand the responsibilities of safety assessors and to identify the user actions during the assessment process. However, the activities in these processes are not specified, especially in the review stage. To make the steps in the review stage explicit, we propose a detailed process flow for the review stage. It is designed for general safety case assessment which is independent on safety case formats. In other words, this process can be applied to both textual and graphical safety cases.

Figure 2.11 shows the detailed steps of our proposed process. There are four key steps in the safety case assessment: 1) prepare for review; 2) validate logic and structure; 3) evaluate quality; 4) record and give feedback. In the first step *Preparation*, the assessor receives a safety case which is developed by safety case developers. Hereby, we assume that the safety case is submitted with additional information wherein the purpose of the safety case and its background are introduced. Before starting the assessment, a number of preparations should be done. The completeness and consistency of the safety case are checked by the assessor. For example, whether there are undeveloped elements. Besides, they also need to check the format of the safety case to select corresponding tool used for the review process.

In the second step *Logic and structure validation*, the assessor should initially review the safety case and determine whether its logic and structure is reasonable and valid. We propose to start reviewing the safety case top down, because it is a natural and logic order of reading GSN-based safety cases.

After each element is reviewed from a logic and inference aspect, the next step *Quality evaluation* is to evaluate the elements from a quality aspect. We suggest to qualify a safety case bottom up, because the most important information is stored in the evidence. In this step, the assessor reviews the content of each evidence document and provides an evaluation for it. Depending on the quality level assigned to the evidence, the assessor can determinate whether the goal supported by the evidence is qualified. In this way, every element in a safety case can be evaluated via its children. Eventually, the top goal can be evaluated based on the quality level of all its branches.

Finally, in the fourth step *Record and feedback*, the assessor needs to record and summarize evaluation results, then gives feedback to developers. Besides, a final judgment can be provided to safety case developers. If the judgment is accepted, then the assessment process finishes. If the judgment is not accepted, the safety case needs to be revised by developers and reviewed again by assessors.



Assessstatus	Accepted
Belief Degree	{(Very Low, 0.0), (Low, 0.1949), (Medium, 0.0), (High, 0.5138), (Very High, 0.0)}
Confidence Percentage	
Description	

Figure 2.12: A screenshot of the AGSN editor

2.5.3.3 The AGSN Editor

To support the proposed process, we have developed an editor (AGSN editor). In the AGSN editor, all assessment steps are supported. Figure 2.12 shows a screenshot of the AGSN editor. For more detailed information of the AGSN editor, we refer to [74].

Firstly, the assessment status is provided to facilitate the validation of logic and structure of a safety case. In Figure 2.12, we could see that in the properties view, the assessment status of the selected element is shown. Secondly, the quality level is designed to facilitate the quality evaluation of safety case elements. Similar to the assessment status, the quality level of an element can also be added or modified via the properties view. To support recording and giving feedback, recommendations, statistical reports and Evidential Reasoning (ER) score calculations are provided. A recommendation can be directly used to give feedback. The statistical report helps the assessors to analyze the assessment and evaluation results, and to provide an overview of the assessment. Finally, the ER algorithm [136], [137] is applied to calculate an overall quality evaluation of a safety case. In Figure 2.12, the degree of belief of the selected element is also shown, which is calculated based on the evaluation results of its evidence. Besides, the ER score of a safety case can also address the uncertainties in evidence. Uncertainties in evidence can affect assessors' confidences in the evaluation process. Showing the confidence degree of an evaluation makes the evaluation more credible and objective.

For more information about this tool, please refer to the full paper on this topic [84].

2.6 Conclusions

As more and more manufacturers in the automotive domain start to comply with the ISO 26262 in their projects, in this chapter we have discussed a number of recent research directions regarding to this standard. We provided a brief introduction of several basic concepts in the standard. Then we discussed safety management parts in details. After that we presented an integrated V model, based on ISO 26262 to emphasize the hierarchical structure of requirements, and to enforce gradual development or refinement of requirements based on higher level requirements and design. Then, we briefly showed a comparison of a number of safety architecture pattens. The results of this comparison can be used as implementation or extension suggestions. Finally, as model-based techniques have been used to support safety assessment, we also described our current research on modeling safety standards, modeling safety cases, and safety case assessment.

Chapter 3

A Holistic Domain Model of ISO 26262 for Model-Based Safety Engineering

The ISO 26262 standard is considered to be the state of the art in safety engineering in the automotive industry. Compliance with this standard is the best practice for achieving safety in the automotive domain. Safety assurance and compliance can be subjective and prone to human error due to the inherent internal inconsistencies and possibly vague requirements. To address this challenge, recent research formalized this domain through model-based engineering. However, most research is focused on the system design aspect of safety, and does not offer a solution for integrating the process aspect in the models. In this chapter, we propose a holistic domain model that can be used for model-based safety engineering. We model both the system design concepts as well as the process elements in the safety domain. To achieve this, we systematically analyze the specification of ISO 26262 and model the concepts. We keep the trace between our model and the standard to resolve the inconsistencies by proposing solutions in the model. Our proposed domain model, supports both the system design and process aspects of safety. Using our proposed model, we provide a specification of the safety lifecycle of ISO 26262 as well as a UML profile for modeling safety artifacts. By integrating the system design and process aspect, we take a step towards automation in this domain. The proposed model may be used for automating both the safety engineering and safety assurance. This automation results in more efficient and less error-prone safety engineering.

3.1 Introduction

The complexity of automotive embedded systems is rising rapidly due to the increasing demand for features and functionality of these systems. This complexity is the result of initiatives such as automated driving and smart mobility. The embedded systems (also known as E/E systems) control the dynamic behavior of the vehicle through actuators thanks to X-by-Wire technologies; this control over behavior makes them safety-related. Organizations and companies face challenges to ensure vehicle safety due to the increasing complexity of these systems.

Functional safety is concerned with safety in case of system failures. Functional safety plays a crucial role in the overall safety as gradually driving tasks and responsibilities are shifted towards the systems. The state of the art in automotive functional safety is captured in ISO/DIS 26262:2018 [54]. Methods for adapting development processes based on this standard have been studied [63], and the industry has shown a high adoption rate. This standard provides guidelines and requirements to ensure functional safety. It views safety from both system design aspect and process aspect. The system design requirements define the safety of the system in various architectural views, for instance, functional and technical views. The specification of these requirements include some general safety concepts such as *fault*, *failure*, *hazard*, as well as more automotive specific ones, such as the *Automotive Safety Integrity Level (ASIL)*. The ASIL ranking classifies the risk of a hazardous event and related safety requirements.

The process aspect defines the safety lifecycle by specifying safety related activities for all development phases and by providing management and organizational requirements. The safety lifecycle ensures a predictable realization of functionally safe systems. As defined by ISO 26262, the safety lifecycle is the entire period that a safety-related system exists, from the concept phase to the decommissioning phase. This standard provides guidelines for performing various safety related activities, including various safety analyses. It indicates work products that are required for or produced by all safety activities. Work products are the documented results (possibly in the form of a model) of a safety activity.

One challenge of achieving compliance to the ISO 26262 standard is that it is subject to expert interpretation. For example, the term ‘hazardous event’ is defined as a “combination of a hazard and an operational situation” in the vocabulary of ISO 26262 (Part 1). Later, in Part 3 of the standard (functional safety during the concept phase), we read: “the operational situation and *operating modes* in which an item’s malfunctioning behavior will result in a hazardous event shall be described $\cdots$”. In this example, it is not clear whether ‘operational situation’ is enough (in combination with a hazard) for defining a ‘hazardous event’ or is an ‘operating mode’ also required.

These types of inconsistencies imply that there is no universally unique interpretation of this standard. The difference in interpretation can cause communication errors, which in turn can become system design errors. These errors have a negative impact on achieving traceability and ensuring safety with the rapid increase of complexity of the systems.

Another compliance challenge originates from the new development processes such as agile or incremental development. These processes entail many updates and changes to the system during the development phase. Safety engineering involves analyzing large quantity of information about the system. Tracing the impact of (in some cases daily) changes to the system on the safety analysis (and consequently on safety assurance) is difficult. For instance, the hazard analysis of a typical Automated Driving (AD) system results in

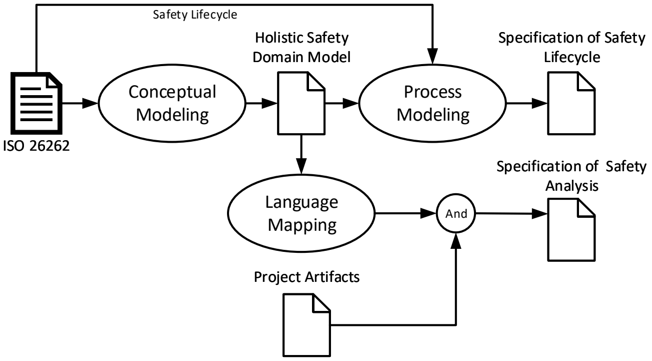


Figure 3.1: Overview our approach for holistic model-based engineering

over a 1000 different variations only during the first step (i.e., functional hazard analysis); each variation includes a function or feature of the AD system, the chosen guide-word for the analysis, justification for applicability of the guideword, the resulting malfunctioning behavior, and finally the possible hazard at the vehicle level. Keeping track of internal consistency of the information over all the variations while performing the analysis is labor-intensive and prone to human errors. The typical human errors do not happen within the content of the analysis, but are about keeping the right information trace. Keeping track of safety issues within the evolutionary approaches to system development (e.g., agile) requires some degree of automation.

The main motivation of this research is to provide a holistic method to address safety from both system design and process aspects. Our solution is based on a formal model that provides the means for integration of functional safety with the other domains such as system engineering and design. Our proposed model integrates the process and system design concerning functional safety. In this chapter, we refer to this model as the Holistic Safety Domain Model (HSDM).

From the system design aspect, we model the primary concepts of specifying a system, as well as the primary concepts of safety analysis. Moreover, we model the relations between these two perspectives. Thus, we can provide a language-agnostic method for the integration of system design with safety analysis.

From the process aspect, our model provides meaningful concepts for changes to the artifacts by modeling activities[1]. These activities enable us to reconstruct the safety lifecycle as proposed by the standard. This way, the definition of the process model goes beyond the document level and expresses the artifacts that are involved in each activity. The result is an unambiguous description of the process, which can be uniquely interpreted in terms of the concepts in the domain model. The unambiguous machine readable specification of the safety lifecycle, may be used for automation of compliance safety engineering.

We show the overview of our approach in this research in Figure 3.1. We create a conceptual model of the safety standard that results in the specification of HSDM. The steps for conceptual modeling are explained in Section 3.3, and the resulting model is described in Section 3.4. We show two different applications of the proposed domain model for: formally specifying the safety lifecycle, as well as a system specification and safety analysis

[1] The primary units of change are the: Create, Read, Update, and Delete (CRUD) actions.

(as an example). We describe the details of the application of HSDM in Section 3.5. We discuss the validity of the proposed model in Section 3.6. Finally we conclude this chapter in Section 3.7.

3.2 Related Work

Conceptual modeling is a way to reduce ambiguities and inconsistencies [123]. There are a few publications regarding model-driven engineering for safety in the literature.

M. Filax *et al.* discuss the applicability of formal notations for increasing the requirement traceability and reliability of systems for the railway domain [29]. They propose a process for formal requirement refinement and verification.

In the avionics domain, models in two different domain-specific modeling languages were used for modeling the functionality, and the safety of a remotely piloted aircraft system. [104] suggests four main information exchange instances in the concept design phase with the focus on the functional architecture. They point out the challenges for ensuring consistency among the model in the two domains (design and safety). Another approach is proposed by A. Legendre *et al.* [73]. They offer a framework by modifying ISO 42010 to integrate the two fields of system architecture and safety engineering.

P. Mauborgne *et al.* published a similar work for the automotive domain [86]. They provide a case study about a fuel delivery system and create a functional safety concept by enriching the functional architecture of the system. F. Franco *et al.* addresses the issue of integration of models from the perspective of OEM-Supplier integration [31]. Similarly, K. Beckers *et al.* discuss the challenge of model-based development with the supply chain of the automotive industry [6]. They provide a UML profile for safety engineering according to ISO 26262, and use a case study to demonstrate their proposed method. However, since they only provide a very high-level description of the process (similar to the safety lifecycle of ISO 26262) to be used for defining the interface between the OEM and supplier, the benefits of a model-based approach (compared to a document-based approach) is not clear. In this chapter, we propose a lifecycle model that specifies the flow on the content level (rather than documents).

A few papers discuss the challenge of integrating models in the automotive domain across different abstraction levels using UML or SysML [25], [116], [117]. R. Weissneggre *et al.* provide a method for integrating the models from UML, MARTE, and SysML [132]. Their method verifies safety requirements by creating simulation tests on the integrated models. They show a successful application in UML, MARTE, and SysML to modeling the system aspect of safety-related automotive systems [133]. K. Thramboulidis *et al.* also proposes SysML profile for capturing the safety related concepts [124]. Similar work is done by D. Cancila *et al.* in [14] where they propose a UML profile to formalize safety-related concepts. In the work of G. Biggs *et al.* in [7], a SysML profile is proposed that can capture the safety-related concerns of a system. Similarly, D. Szymanski *et al.* [122] also show a case on using SysML extension for modeling functional safety artifacts; moreover, they provide tool support for integration of Enterprise Architect and Matlab through automatic code generation. P. Mauborgne *et al.* in [85] give a conceptual model for hazard analysis and provide a methodology for operational and system risk analysis. However, the mentioned profiles, or languages lacked the ability to capture the process aspect of safety engineering. Our proposed domain model captures the process as well as the system design aspects.

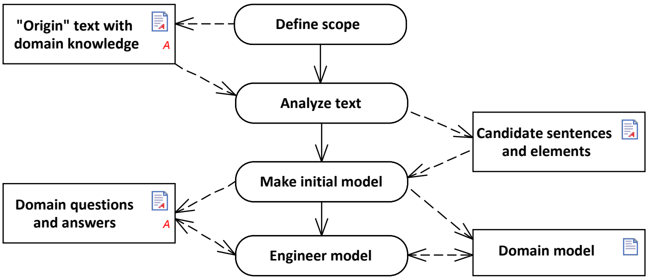

"Origin" text with domain knowledge
Define scope
Analyze text
Candidate sentences and elements
Make initial model
Domain questions and answers
Engineer model
Domain model

Figure 3.2: Overview of our adaptation of the KISS method

Y. Luo gives a comprehensive thesis on the application of conceptual modeling for safety assurance in the automotive domain [76]. In [84] they offer a formal process for safety case assessment as well as tool support for their proposed process. Model-based methods have also been applied to other phases of a safety-related product. An example is modeling product lines from safety and variability viewpoints by A. Salikiryaki *et al.* in [115]. In this chapter, we focus on modeling the conceptual phase of safety engineering from a development perspective (as opposed to assurance). Our domain model can be used for specifying a safety-related system and the safety analysis related to that system.

3.3 Methodology

In this research, inspired by the Kristen Informatie and Software Services (KISS) method for object orientation [71] we create a conceptual model method. The KISS method offers a procedure to process natural language and is intended to be used for object-oriented modeling. We apply our adaptation of this method to the safety domain specified in ISO 26262. Therefore, the resulting object models and behavior models represent, respectively, the system design concepts and processes. These models define the semantics of the domain of ISO 26262 safety engineering. We use UML as the notation language of our models [110].

3.3.1 Method Overview

Our adaptation of the KISS method has four main steps: 1. define the scope, 2. analyze text, 3. make the initial model, 4. engineer the model. An overview of these steps is shown in Figure 3.2.

In the first step, the objective is to define the boundaries of the target domain and the purpose of the model. To define the scope, we need to choose an *"origin"* text that is used as the input for modeling.

In the next step, we analyze the "origin" text and convert it from an unrestricted text (in natural language) into a set of structured sentences and candidate model elements. The "origin" text can be in any format; it should contain enough information of the domain, and the target users of the domain model should have consensus on the content.

An initial model of the domain is created by determining the relevance of each extracted sentence, identifying candidate model elements, and eliminating homonyms and synonyms. We create the model by converting specific *verbs* into UML activities, and specific *nouns* into UML classes. In this step two views are created:

Interaction view: that captures the domain activities and their participating domain classes. In our case, these activities refer to safety activities within the development lifecycle, e.g., identifying hazards or determining safety goals (during HARA). The order of the execution are not given in this view; however, we use the declared activities from this view to specify the safety lifecycle (which captures the expected order of execution). The details of the safety lifecycle are described in Subsection 3.5.1.

It should be noted that (due to the choice of modeling language) we use activity diagrams for specifying both the safety development processes as well as the behavior of the system of interest. While this may seem confusing, we deem the risk of misunderstanding to be minimal as these two models have very different use and are not likely to be used in the same context.

Static view: that shows all the static relations between the domain classes. Most relations between classes are a result of a domain activity in which they participated. The static view also shows the subset relations between the domain classes in the form of specializations. This view is the basis for creating a profile for safety specification of a system. The profile and an example application is described in detail in Subsection 3.5.2.

Optionally, there could be a third view:

Attribute view: that shows the attributes of domain classes and parameters of the activities.

Furthermore, it is possible to divide the model into sub-packages to manage the models. These packages could correspond to the significant part of the target domain. For example, in our model, we have a package for Item Definition, and another for Hazard Analysis and Risk Assessment. These happen to be subsections of Part 3 of ISO 26262 as well as important steps in the safety lifecycle.

Finally, in the last step, the domain model is refined through an iterative process that ensures the correctness and the consistency of the model. As an example, we check that all the domain classes have an instantiating action; and that each relation in the static view is related to an action from the interaction view.

3.3.2 Application of the Method for HSDM

Here we describe the mentioned steps for application of HSDM. For the first step, we define the scope of HSDM to be Part 3 of ISO 26262 (excluding the Functional Safety Concept Section) in addition to the relevant clauses in Parts 1, 8 and 10.

The intended purpose of HSDM is twofold: first, to specify the safety lifecycle as described in ISO 26262; second, to specify safety work products using models. The content of HSDM reflects the authors' interpretation of this standard. The interpretation is based on the authors' knowledge and expertise in the fields of functional safety and modeling.

Next, we analyzed the text of this standard to extract the candidate elements of the domain model. Table 3.1 shows a few examples of extracted sentences. In this step, the vocabulary of the standard (Part 1) proved useful for identifying the concepts of the domain model. However, the vocabulary cannot be the only source for the domain model due to a few limitations. The primary limitation of the vocabulary is that it does not give any information regarding the process aspect. This is expected since a process is based on the relation between the terms and concepts and can not be expressed by only defining the terms.

Another limitation is that the vocabulary may be over or under specified, which means that there could be important concepts missing in the vocabulary or on the other hand

Table 3.1: Example of sentence extraction

Original sentence	#	Extracted sentence/phrase
Part 1: 3.74 hazardous event: combination of a hazard and an operational situation	1	Hazardous event is a combination.
	2	Hazardous event has hazard.
	3	Hazardous event has operational situation.
Part 3: 6.4.2.1 The operational situations and operating modes in which an item's malfunctioning behavior will result in a hazardous event shall be described, both for cases when the vehicle is correctly used and when it is incorrectly used in a foreseeable way.	4	Item malfunctioning behavior results in hazardous event.
	5	To describe operational situation.
	6	To describe operating mode.
	7	To use vehicle in a correct or foreseeable incorrect way.
	8	To use involves correctness.
Part 3: 6.4.2.3 Hazards caused by malfunctioning behavior of the item shall be defined at the vehicle level	9	Item malfunctioning behavior causes hazard.
	10	To define hazards caused by Item malfunctioning behavior.
Part 3: 6.4.2.5 Relevant hazardous events shall be determined	11	To determine a hazardous event.

concepts could be included that are not crucial. An example of this limitation can be found in the terms related to estimating the severity part of the hazard analysis in Part 3 of ISO 26262. The term "hazard consequence" is missing in the vocabulary of the ISO 26262; instead, we find the definition of "harm". This makes sense from the perspective of the definitions, as *harm* should be the basis for estimating the severity. However, the body of the standard suggests (also in practice we see the same) that the severity should be based on "hazard consequence". In this case, "hazard consequence" is the accident that may happen as a result of a "hazardous event", and "harm" would be the outcome of that accident.

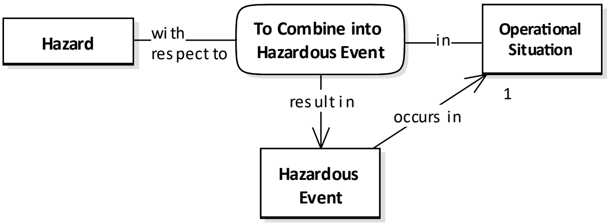
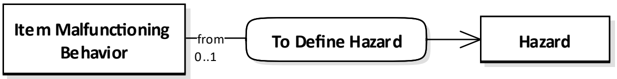
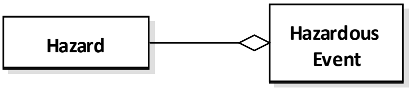
Moreover, the use of the terms in the vocabulary and the body may not be consistent in all cases. An example of inconsistency was mentioned in the introduction of this chapter. To give another example: we read a slightly different connotation in the extracted sentences number 4 and 9:

"Item malfunctioning behavior [results in hazardous event] ***or*** [causes hazard]."

Here, the inconsistency is in the assumed causal relation between a malfunctioning behavior, hazard, and hazardous event. The first option (in our view incorrectly) assumes a causal relation between a malfunctioning behavior and a hazardous event; whereas, the second option (correctly) suggests a causal relation from a malfunctioning behavior to a hazard.

In the next step, we make an initial model of the domain. Table 3.2 shows a few examples of sentences and their representation in the initial model. The initial model is the starting point for the model engineering phase. The result of this step is a domain model that matches the defined scope, which is described in details in the next section.

Table 3.2: Example of sentence initial model

#	View type	Model representation
1 + 11	Interaction view	 Hazard with respect to To Combine into Hazardous Event in Operational Situation 1 result in occurs in Hazardous Event
9 + 10	Interaction view	 Item Malfunctioning Behavior from 0..1 To Define Hazard Hazard
2	Static view	

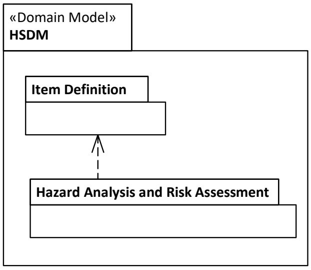


Figure 3.3: The Holistic Safety Domain Model packages. We show the dependency of the sub-packages in this diagram.

3.4 The Holistic Safety Domain Model

We modeled HSDM in three UML packages (including two sub-packages). The top-level package (also named HSDM) contains the more generic concepts that are applicable for all sub-packages. The Item Definition package introduces the concepts and elements related to define the Item. Similarly, needed concepts hazard analysis and risk assessment are in the related package. An overview of the model is shown by the use of a package diagram in Figure 3.3.

In this section, we describe the content of each package in detail. We start with the description of HSDM package that contains the more generic concepts; then we describe the Item Definition, and finally the Hazard Analysis and Risk Assessment package.

3.4.1 HSDM

We introduce a few abstract concepts to group the various concepts that are used for each safety analysis. We use these abstract concepts to generalize the domain concepts and assign shared characteristics to them.

3.4.1.1 HSDM – Static Model

In this package we have two abstract classes: Safety Analysis Object and ASIL Object. The only other abstract class in HSDM is Safety Design Object, which is defined in the Item Definition sub-package. In addition, we have an abstract activity: Safety Analysis Action as described in the following subsection.

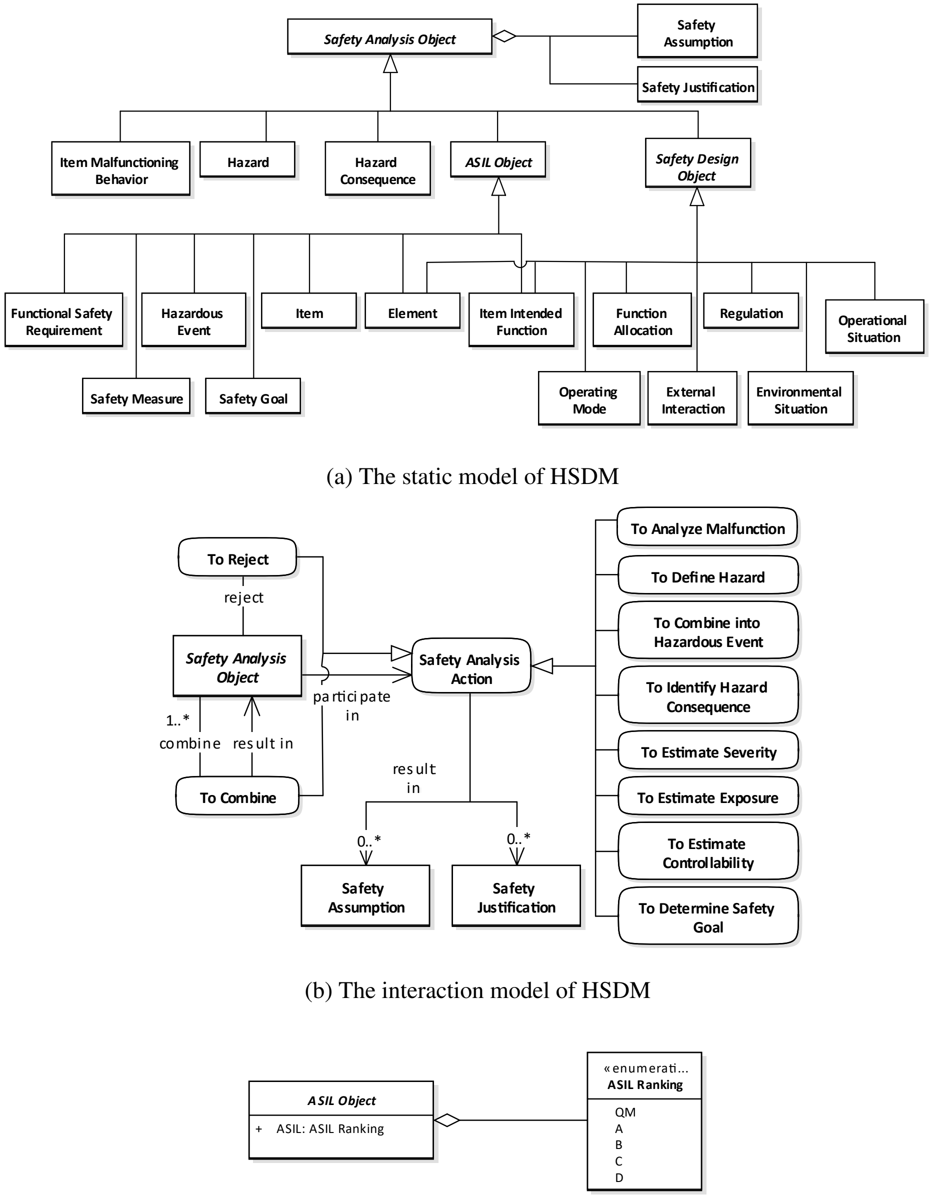
Safety Analysis Object is the most generic class in HSDM and refers to the objects that are created or used during any safety analysis. A Safety Analysis Object has Safety Assumptions as well as Safety Justifications. Safety Assumption refer to all the assumptions made during a Safety Analysis Action, and Safety Justification refer to the justification used during those actions. An Item Malfunctioning Behavior and a Hazard Consequence are Safety Analysis Objects. Moreover, ASIL Object and Safety Design Object are also Safety Analysis Objects. An ASIL Object refers to those objects that can be assigned an ASIL ranking. Note that the ASIL rankings are A to D, however due to practical reasons we added QM as part of the possible ASIL rankings in this model. A few concepts inherit the ASIL ranking; the classes: Hazardous Event, Safety Goal, Item, Functional Safety Requirement, Safety Measure and Element are all ASIL Objects. Safety Design Object are classes that are used in defining an Item. This is explained in more detail in Section 3.4.2. Figure 3.4a shows an overview of these concepts.

3.4.1.2 HSDM – Interaction Model

The diagram in Figure 3.4b shows the activities for safety analysis. During a safety analysis, Safety Analysis Objects undergo various Safety Analysis Actions, which results in their creation or change. There are two “editorial” shared activities between all safety analyses: To Combine, and To Reject. A few Safety Analysis Objects can be combined into another Safety Analysis Object. A Safety Analysis Object can be rejected that means it is not considered for the remainder of safety analyses, but the Assumptions and Justifications need to be documented. Analyzing Malfunction, Determining Hazard, Identifying Hazardous Event, Identifying Hazard Consequence, Estimating Severity, Estimating Exposure, Estimating Controllability, and Determining Safety Goal are all Safety Analysis Actions. These activities are shown in Figure 3.4b.

3.4.1.3 HSDM – Attribute Model

The diagram in Figure 3.4c shows the concepts that are generic for safety ranking. An ASIL Object is ranked at an ASIL Ranking. ASIL Ranking is one of five levels to specify the requirements of ISO 26262 and safety measures for avoiding an unreasonable risk, with D representing the most stringent and A the least critical, and QM indicating that no safety action is required.

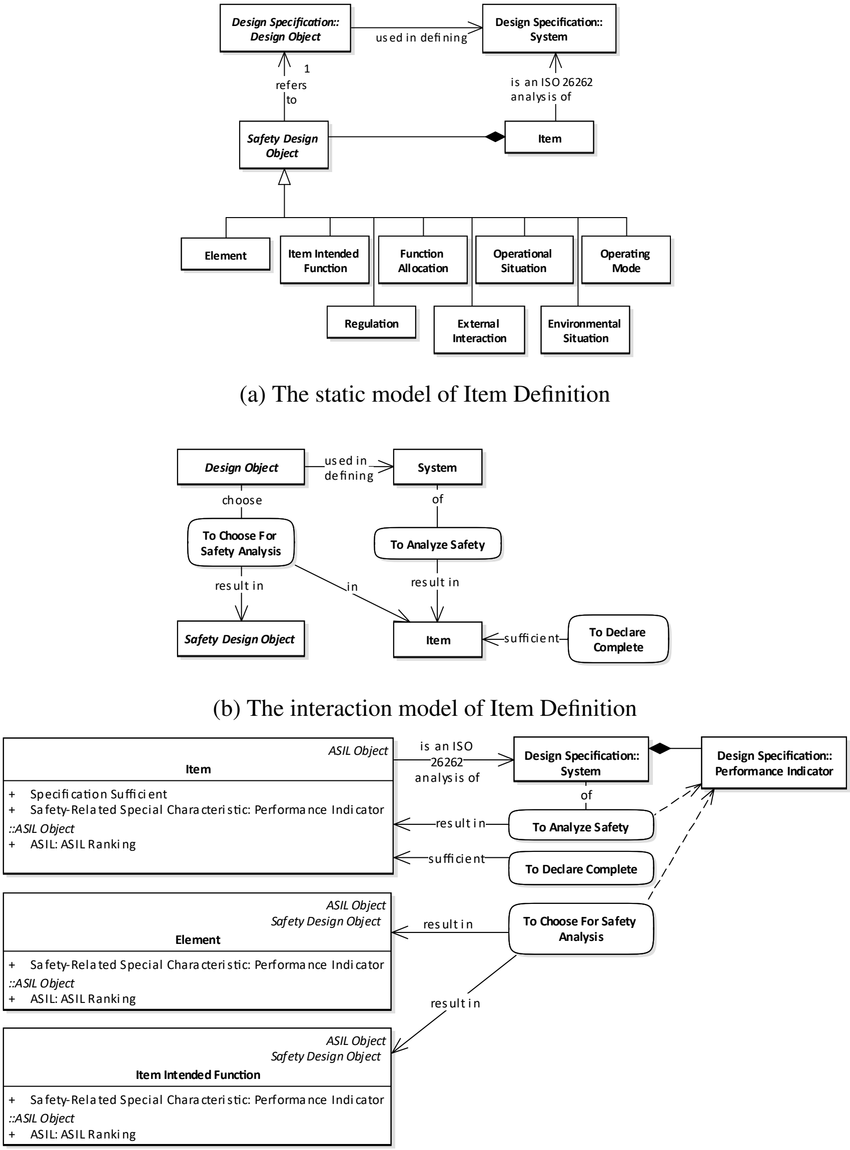


(a) The static model of HSDM

(b) The interaction model of HSDM

(c) The attribute model of HSDM

Figure 3.4: The models of HSDM



(a) The static model of Item Definition

(b) The interaction model of Item Definition

(c) The attribute model of Item Definition

Figure 3.5: The Item Definition Models

3.4.2 Item Definition

This subsection contains the diagrams that specify concepts related to the Item. Note that we provide a concrete example of the concepts described here in Subsection 3.5.2.

3.4.2.1 Item Definition – Static Model

Figure 3.5a shows the concepts that are required to define an Item. Here, we assume that a set of *Design Objects* are used for defining a System. Given this assumption, the Item is the system that is analyzed according to the ISO 26262 standard and the structure of the Item is based on the structure of a System. Typically, the Item reflects the same specification of the System of interest. Each type of Safety Design Object is based on a Design Object with a similar type. As shown in Figure 3.4a, Safety Design Objects can be any of the following: Element, Item Intended Function, Function Allocation, Operating Mode, Regulation, External Interaction, Environmental Situation, or Operational Situation.

3.4.2.2 Item Definition – Interaction Model

A System can be analyzed for safety which results in an Item. Design Objects can be Chosen For Safety Analysis of an Item, which results in a Safety Design Object. The Item can be declared complete such that the rest of the safety activities can be started. The diagram in Figure 3.5b shows the activities that make up the definition of Item.

3.4.2.3 Item Definition – Attribute Model

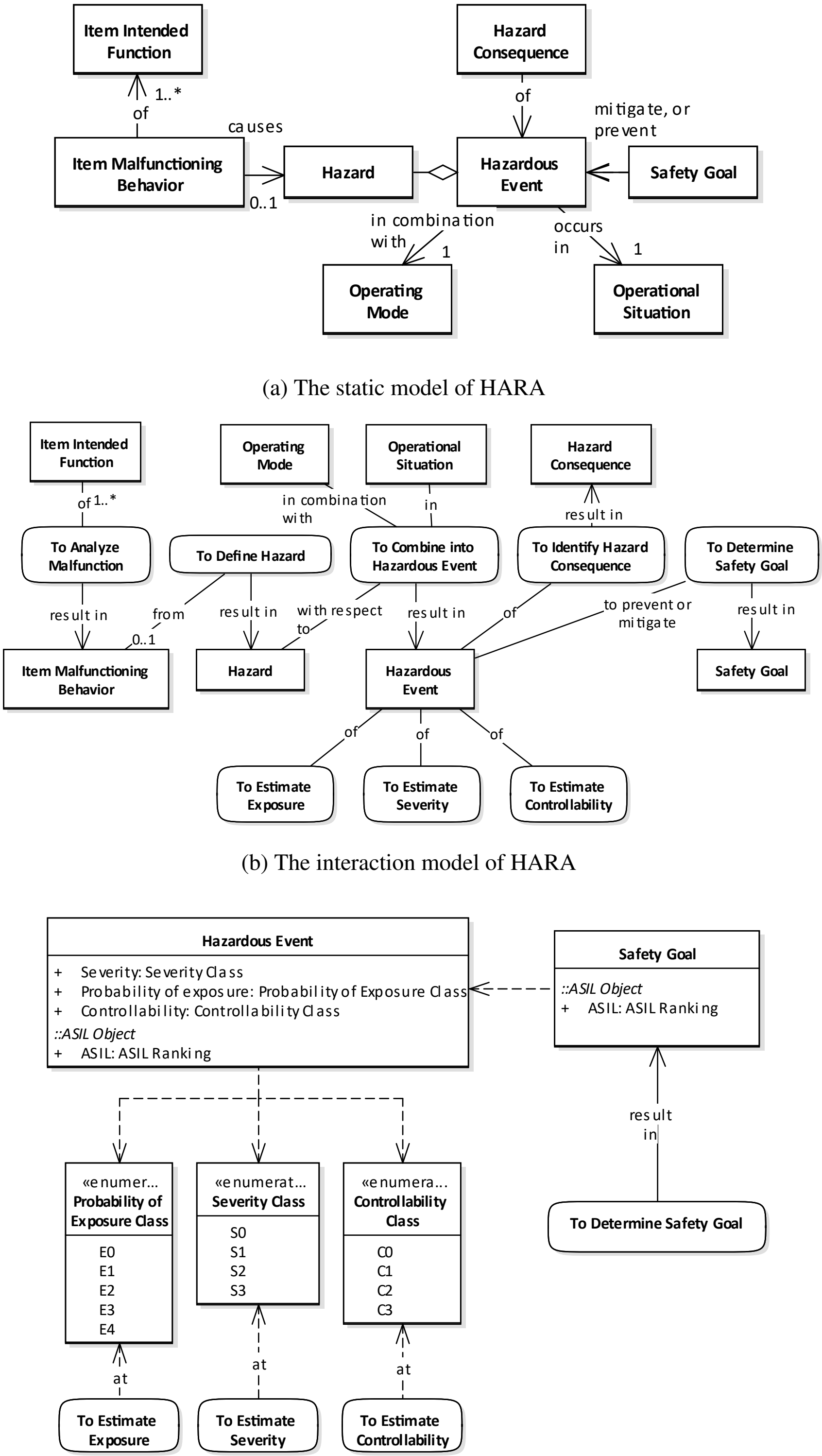
The attributes of the concepts related to Item are shown in Figure 3.5c. Item, Element, and Item Intended Function are ASIL Objects and have ASIL as an attribute. Moreover, Item, Element and Item Intended Function have a safety specific attribute, i.e Safety-Related Special Characteristic, which are the special system characteristics that can be used as an measured indicator of functional safety. These characteristics should be based on the Performance Indicators of the System, assuming that the performance indicators of the system are known. Otherwise, the Safety-Related Special Characteristic need to be defined during Item Definition. Finally, Item has Specification Sufficient, which indicates that the Item Definition has sufficient information for the safety activities to start.

3.4.3 Hazard Analysis and Risk Assessment

Here we describe all the concepts and actions required for performing Hazard Analysis and Risk Assessment (HARA) based on ISO 26262.

3.4.3.1 HARA – Static Model

Figure 3.6a shows classes required for HARA. Item Malfunctioning Behavior is malfunction of one or more Item Intended Functions. Please note that in the ISO 26262, the phrase “Malfunctioning behavior” is used for failure or unintended behavior of an Item, but we use Item Malfunctioning Behavior to avoid possible misunderstandings. An Item Malfunctioning Behavior causes Hazards. Hazards in combination with an Operating Mode may occur in an Operational Situation, which is specified in a Hazardous Event. Note that in the vocabulary of ISO 26262, Hazardous Event is a combination of Hazard and Operational Situation, but in Part 3 of the standard (functional safety during the concept phase) Operating Mode is also referenced for specification of Hazardous event. Therefore, in our model we also consider Operating Mode to maximally capture the specifications of ISO 26262. Hazard Consequences of a Hazardous Event should



(a) The static model of HARA

(b) The interaction model of HARA

(c) The attribute model of HARA

Figure 3.6: The models of HARA

be identified. Hazard Consequence is not defined in ISO 26262. We suggest to define it as a direct result of an Hazardous Event. For example, head-on collision or leaving lane could be instances of Hazard Consequence. A Safety Goal mitigates or prevents Hazardous Events.

3.4.3.2 HARA – Interaction Model

The diagram in Figure 3.6b shows the activities in the hazard analysis package. Analyzing Malfunctioning of one or more Item Intended Functions results in the identification of zero or one Item Malfunctions Behaviors. Hazards are Determined from Item Malfunctioning Behavior. Hazard Consequences of Hazardous Events are Identified. The Probability of exposure of a Hazardous Event, as well as its Severity and Controllability are Estimated and ranked. Finally, a Safety Goal is Determined to prevent or mitigate the Hazardous Event.

3.4.3.3 HARA – Attribute Model

A Hazardous Event has Controllability, Probability of exposure, and Severity that are classified respectively at Controllability Class, Probability of Exposure Class and Severity Class. The three different estimation activities have a parameter to capture their specific estimation aspects. The aggregation of these three parameters determines the ASIL of a Hazardous Event. Subsequently, the ASIL of a Safety Goal is dependent (is the maximum) on the relevant Hazardous Events.

3.5 Application of HSDM

In this section we discuss how the HSDM is used. We provide two applications of HSDM. The first application is to (partially) specify the safety lifecycle of ISO 26262. This corresponds to the Safety Lifecycle package shown in Figure 3.7. We formally specify the Item Definition and HARA. For each process, we use an activity diagram to express the flow and the involved artifacts. We ensure that the actions and objects only use domain model classes and activities.

Furthermore, we use our domain model for specifying an example ACC system. We provide a UML profile that can be used for modeling safety analysis.

3.5.1 Specification of the Safety Lifecycle

The safety lifecycle of ISO 26262 captures all the required safety activities during all phases of a vehicle life cycle. The workflow of this lifecycle is shown in Figure 3.7. The scope of this chapter is limited to the Item Definition, and Hazard Analysis and Risk Assessment activities. In this subsection, we introduce a formal definition of these two activities based on HSDM. We use UML for this definition, as that is the language in which we defined HSDM.

We begin with modeling the top-level process elements similar to the specified safety lifecycle of ISO 26262. Then, we refine these high-level processes using the atomic actions from HSDM. The top-level process is shown in Figure 3.8.

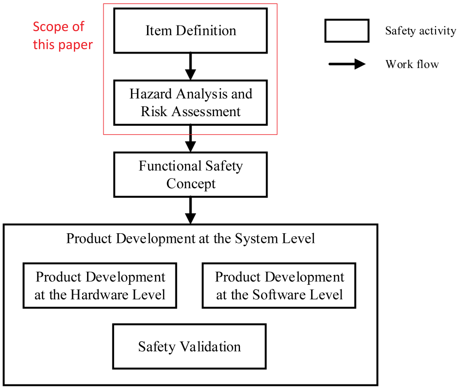

Scope of this paper
Item Definition
Hazard Analysis and Risk Assessment
Functional Safety Concept
Product Development at the System Level
Product Development at the Hardware Level
Product Development at the Software Level
Safety Validation
Safety activity
Work flow

Figure 3.7: Simplified safety lifecycle during concept and development phases recommended by ISO/DIS 26262:2018

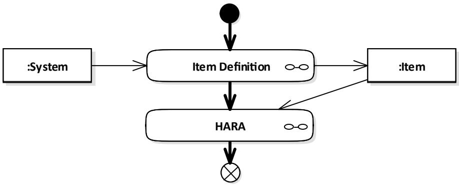


Figure 3.8: Overview of the portion of Safety lifecycle within the scope of this chapter

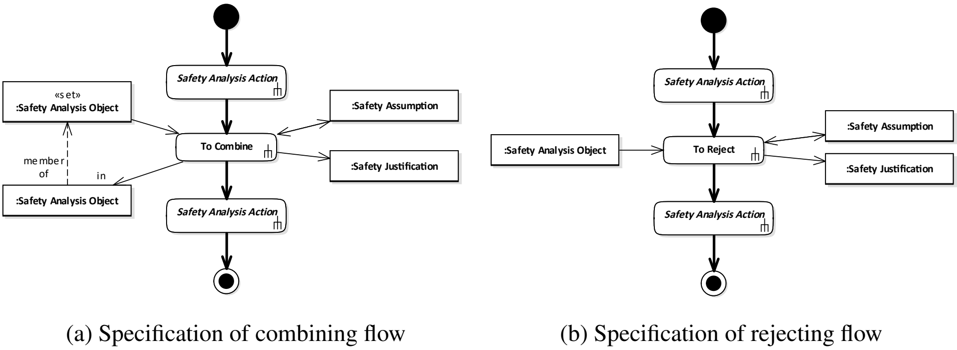
3.5.1.1 The Generic Flows

Benefiting from the generalized classes and actions of HSDM, we can draw some of the generic flows. These flows are related to the "bookkeeping" part of safety analysis and are possible after any safety analysis action. Note that all the activity flows specified here are a type of Safety Analysis Activity, which means that they potentially result in Safety Justifications and/or Safety Assumptions. Safety Analysis Actions are shown in Figure 3.4b include: Analyzing Malfunction, Determining Hazard, Identifying Hazardous Event, Identifying Hazard Consequence, Estimating Severity, Estimating Exposure, Estimating Controllability, and Determining Safety Goal.

The first generic flow, as shown in Figure 3.9a, describes the actions for combining a set of Safety Analysis Objects into one. The combination results in possible Safety Assumptions and Safety Justifications. For example, similar Safety Goals that address the same functionality may be *combined* into one Safety Goal.

The second generic flow is shown in Figure 3.9b. This flow allows to disregard any Safety Analysis Object for the remainder of the safety lifecycle after recording the assumptions and justifications for doing so. For example, during hazard identification, if a Hazard and Operability (HAZOP) guide-word[2] is not applicable to a function, the combination is *rejected* and not considered for the next analysis step.

[2] HAZOP guide-worlds such as "None", "Too much", "Too late" are used to identify possible malfunctions of the intended functions



(a) Specification of combining flow

(b) Specification of rejecting flow

Figure 3.9: Specification of generic flows

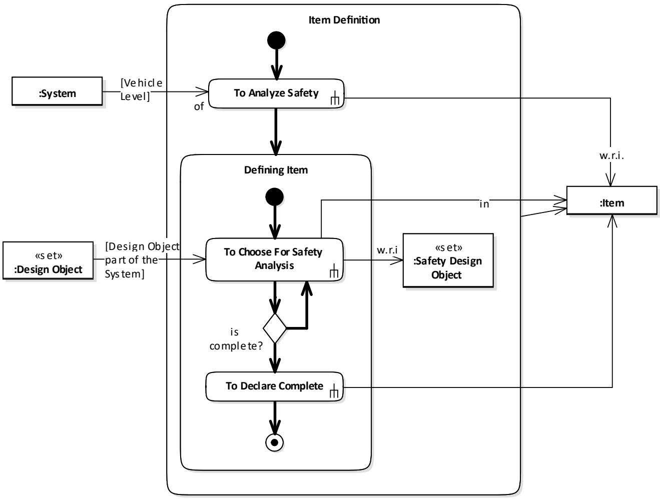


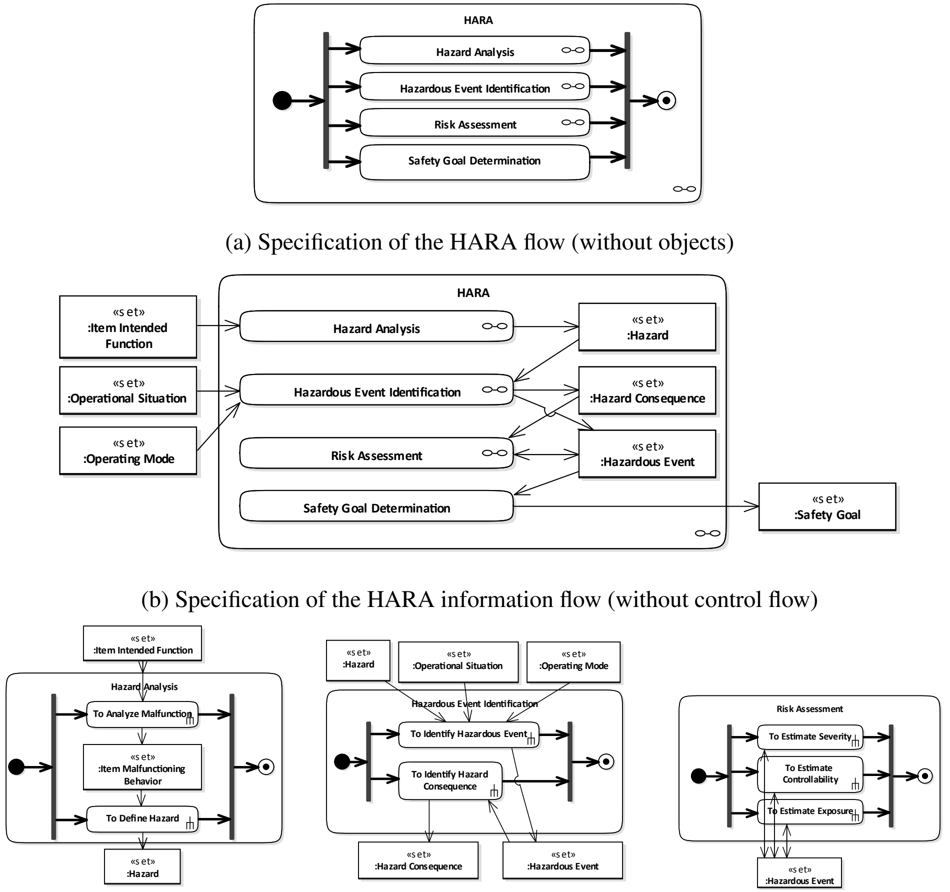
Figure 3.10: Specification of the Item Definition flow

3.5.1.2 Item Definition Flow

The diagram in Figure 3.10 shows the flow of actions for defining an Item. The first step is to select a vehicle level System to be analyzed for safety; this results in the creation of the Item. The second recurring action is choosing a Design Object from the selected System to be part of the safety analysis of the Item. This activity results in the creation of Safety Design Objects that are part of the Item. The final activity is deciding whether the Item contains enough information to continue with the rest of safety lifecycle.

3.5.1.3 Hazard Analysis and Risk Assessment Flow

Figure 3.11a shows the flow for Hazard Analysis and Risk Assessment (HARA). There are four main steps: Hazard Analysis, Hazardous Event Identification, Risk Assessment, and Safety Goal Determination, which are shown by the corresponding activities in this diagram. These steps are further defined in terms of actions of HSDM.


HARA
Hazard Analysis
Hazardous Event Identification
Risk Assessment
Safety Goal Determination
«set» :Item Intended Function
«set» :Operational Situation
«set» :Operating Mode
«set» :Hazard
«set» :Hazard Consequence
«set» :Hazardous Event
«set» :Safety Goal
To Analyze Malfunction
«set» :Item Malfunctioning Behavior
To Define Hazard
To Identify Hazardous Event
To Identify Hazard Consequence
To Estimate Severity
To Estimate Controllability
To Estimate Exposure

(a) Specification of the HARA flow (without objects)

(b) Specification of the HARA information flow (without control flow)

(c) Specification of Hazard Analysis

(d) Specification of Hazardous Event Identification

(e) Specification of Risk Assessment

Figure 3.11: Specification of the HARA flow

Depending on availability of information or the stage of HARA, a safety engineer may choose to perform either of these steps. The information flow is shown in Figure 3.11b.

In Hazard Analysis, depending on availability of information, a safety engineer may perform either Malfunction Analysis or Hazard Determination. To Analyze Malfunction each Item Intended Function (from the set of all Item Intended Functions) is analyzed to identify potential Item Malfunctioning Behavior. To Determine Hazards, each Item Malfunctioning Behavior (from the set of Item Malfunctioning Behaviors) is considered to identify potential Hazards on the vehicle level. These actions are shown in Figure 3.11c.

As shown in Figure 3.11d, the next step is Hazardous Event Identification. Similar to the previous step, depending on availability of information either of the two activities could be performed: To Identify Hazardous Events or To Identify Hazard Consequence. To Identify Hazardous Events, each Hazard is considered in combination with Operational Situation and in an Operating Mode. To Identify Hazard Consequence, each Haz-

Table 3.3: Mapping of HSDM objects for UML profile

Meta Class	HSDM Class
Class	Item, Element, Hazard, Hazardous Event, Hazard Consequence
Activity	Item Intended Function, Item Malfunctioning Behavior
Requirement	Regulation, Safety Goal
State	Operating Mode
Interface	External Interaction
AssociationClass	Function Allocation
UseCase	Environmental Situation, Operational Situation

ardous Event is analyzed to identify its consequence (at vehicle level).

In the third step (shown in Figure 3.11e), each Hazardous Event is classified based on its associated risk. There are three activities in this step: To Estimate Severity, To Estimate Controllability, and To Estimate Exposure. Each activity results in determination of the corresponding component of ASIL of the Hazardous Event.

The last step is determining the Safety Goal that prevents or mitigates the associated Hazardous Event (only those with an ASIL A or higher need a Safety Goal).

3.5.2 Safety Specification Example

Here, we give an example of how HSDM could be used for expressing safety information of an automotive system. First, we describe an UML profile based on the HSDM that enables modeling the safety concepts according to HSDM. Then we use the profile to model an example system. We use an Adaptive Cruise Control (ACC) system as our example. ACC systems are relatively well known and there is sufficient publicly available information to describe their functionality. Moreover, the ACC functions are interesting enough as demonstration of hazard analysis and are not too complex to divert the attention of the reader.

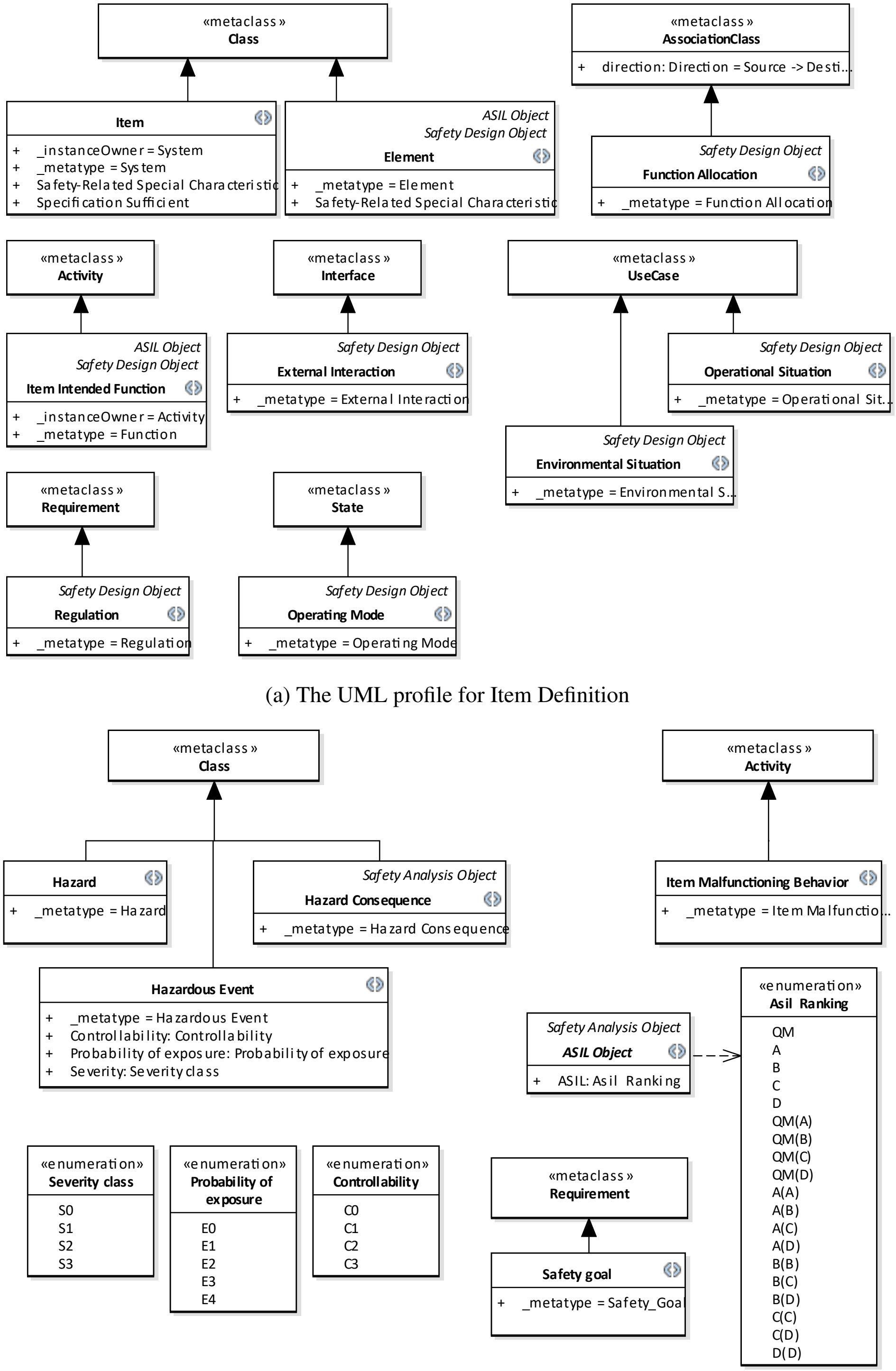
3.5.2.1 The Safety Profile

The UML profiles for modeling an Item and HARA are shown in Figure 3.12. To make the profiles we use the static models of HSDM as illustrated. We give an overview of the choices for metaclasses in Table 3.3.

3.5.2.2 ACC Example

We use the description of an ACC system from [21] to model the system:

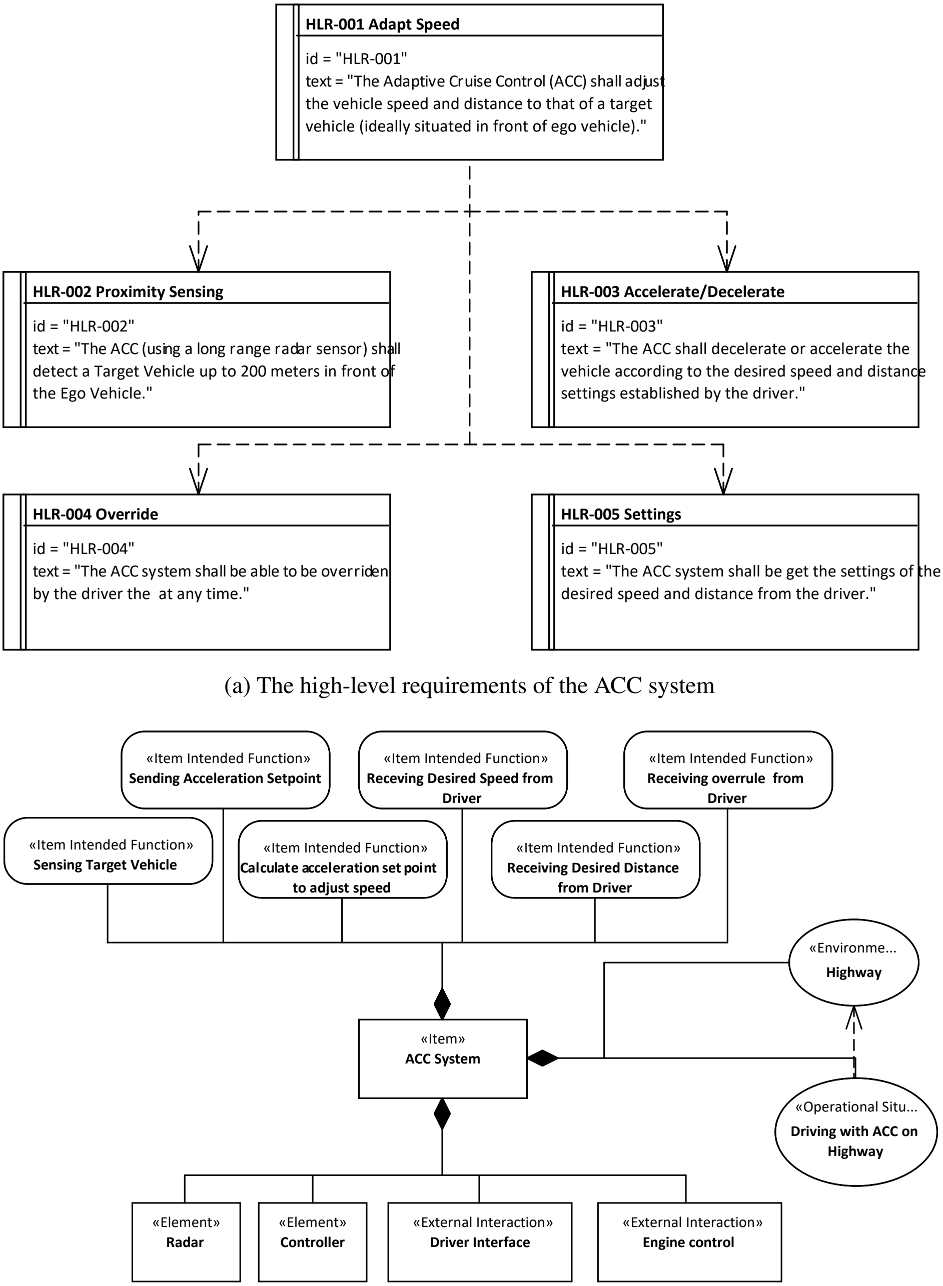
> “TRW’s Adaptive Cruise Control (ACC) technology improves upon standard cruise control by automatically adjusting the vehicle speed and distance to that of a target vehicle. ACC uses a long range radar sensor to detect a target



(a) The UML profile for Item Definition

(b) The UML profile for HARA

Figure 3.12: The UML profile of HSDM



HLR-001 Adapt Speed

id = "HLR-001"
text = "The Adaptive Cruise Control (ACC) shall adjust the vehicle speed and distance to that of a target vehicle (ideally situated in front of ego vehicle)."

HLR-002 Proximity Sensing

id = "HLR-002"
text = "The ACC (using a long range radar sensor) shall detect a Target Vehicle up to 200 meters in front of the Ego Vehicle."

HLR-003 Accelerate/Decelerate

id = "HLR-003"
text = "The ACC shall decelerate or accelerate the vehicle according to the desired speed and distance settings established by the driver."

HLR-004 Override

id = "HLR-004"
text = "The ACC system shall be able to be overriden by the driver the at any time."

HLR-005 Settings

id = "HLR-005"
text = "The ACC system shall be get the settings of the desired speed and distance from the driver."

(a) The high-level requirements of the ACC system

«Item Intended Function» Sending Acceleration Setpoint
«Item Intended Function» Receving Desired Speed from Driver
«Item Intended Function» Receiving overrule from Driver
«Item Intended Function» Sensing Target Vehicle
«Item Intended Function» Calculate acceleration set point to adjust speed
«Item Intended Function» Receiving Desired Distance from Driver
«Environme... Highway
«Item» ACC System
«Operational Situ... Driving with ACC on Highway
«Element» Radar
«Element» Controller
«External Interaction» Driver Interface
«External Interaction» Engine control

(b) The Item Definition of the ACC System

Figure 3.13: The model specification of the ACC System

vehicle up to 200 meters in front and automatically adjusts the ACC vehicle speed and gap accordingly. ACC decelerates or accelerates the vehicle accord-

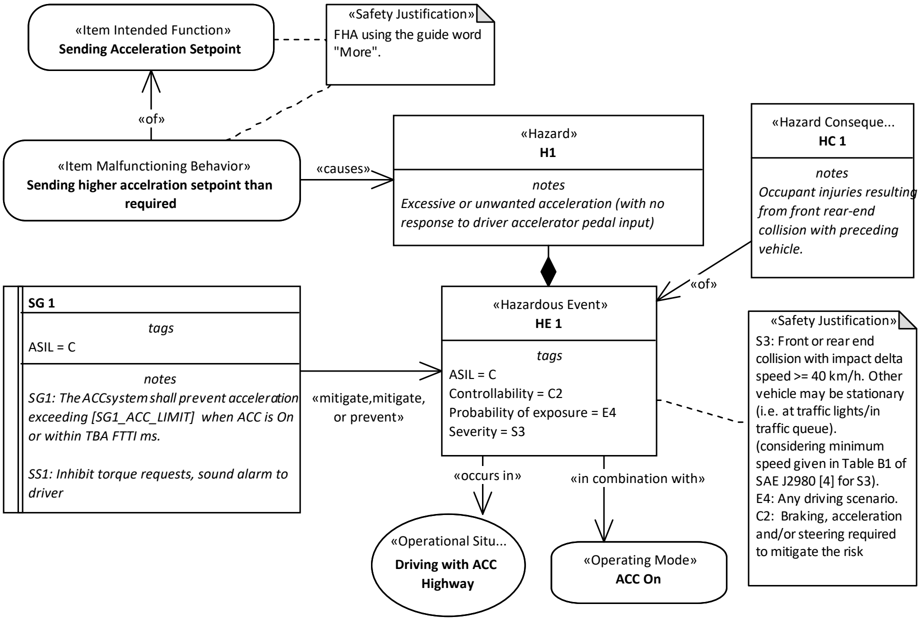

«Item Intended Function»
Sending Acceleration Setpoint
«Safety Justification»
FHA using the guide word "More".
«of»
«Item Malfunctioning Behavior»
Sending higher accelration setpoint than required
«causes»
«Hazard»
H1
notes
Excessive or unwanted acceleration (with no response to driver accelerator pedal input)
«Hazard Conseque...
HC 1
notes
Occupant injuries resultin
from front rear-end collision with preceding vehicle.
«of»
SG 1
tags
ASIL = C
notes
SG1: The ACCsystem shall prevent acceleration exceeding [SG1_ACC_LIMIT] when ACC is On or within TBA FTTI ms.
SS1: Inhibit torque requests, sound alarm to driver
«mitigate,mitigate, or prevent»
«Hazardous Event»
HE 1
tags
ASIL = C
Controllability = C2
Probability of exposure = E4
Severity = S3
«Safety Justification»
S3: Front or rear end collision with impact delta speed >= 40 km/h. Other vehicle may be stationary (i.e. at traffic lights/in traffic queue). (considering minimum speed given in Table B1 of SAE J2980 [4] for S3).
E4: Any driving scenario.
C2: Braking, acceleration and/or steering required to mitigate the risk
«occurs in»
«in combination with»
«Operational Situ...
Driving with ACC Highway
«Operating Mode»
ACC On

Figure 3.14: Example of the HARA of the ACC System

> ing to the desired speed and distance settings established by the driver. As per standard cruise control, the driver can override the system at any time".

Based on the system description, we derive five high-level requirements as shown in Figure 3.13a. As we show, the first requirement states the ability of adjusting the speed depending on the target vehicle. This requirement depends on the other four requirements on the sensing conditions and range, control of acceleration, as well as the settings and override by the driver.

The high-level requirements can be the basis for the Item Definition. Based on the requirement specification, the ACC system is decomposed into two main elements (Radar and Controller) as well as two interface blocks for interacting with the driver and the Engine Control. We inferred six main functionalities of the ACC system: Sensing the target vehicle, Calculating the acceleration setpoint, Sending the acceleration set point, and three other functions that provide the required interaction with the driver for settings and override. As the ACC system is an improvement over the standard cruise control, we inferred that the target operational domain is the highway. The diagram in Figure 3.13b shows the model of the ACC system using the introduced UML profile. Note that we left the function allocations out of this diagram to keep it readable. We assume the allocation of the functions to elements to be straight forward given the few number of elements.

Using the information provided in the Item Definition of the ACC system, we perform the Hazard Analysis and Risk Assessment. The model in Figure 3.14 shows an example of such an analysis. In this example, we show the analysis on the Intended Function "Sending Acceleration Setpoint." Using Functional Hazard Analysis (FHA) method and guide word "more," we identify a potential Malfunctioning Behavior of the system for sending too high acceleration request to the Engine. This malfunctioning behavior may cause a Hazard of excessive acceleration. This hazard, in combination with the Operational Situation and the Operating Mode constructs the Hazardous Event “HE 1”. The Consequence of this Hazardous Event could be a rear-end collision and therefore injuries to the occupants. We assess the risk of this Hazardous Event at ASIL C and provide the Justification for such ranking. Finally, we determine the Safety Goal “SG 1” for mitigating the impact of such hazard by limiting the maximum allowed acceleration of the ACC system.

3.6 Validation

In this section we discuss the threats to the validity of our proposed models and the mitigation measures we used in this research.

The most important threat to the validity of HSDM is wrong interpretations of the ISO 26262 specification during conceptual modeling (see Figure 3.1). As discussed in Section 3.3, even though the modeling method ensures traceability to the specifications of the original text (ISO 26262), our interpretations of the ISO 26262 text have an impact on modeling decisions such as the choice for capturing the classes, relations, and activities. The impact of this threat is that the HSDM model results in superfluous models that are not suitable for application and not agreeable by the concerned communities.

We have taken mitigative measures against this threat on two levels. On the applicability level, we showed that the model can be used for modeling the safety related work flows, and therefore reconstructing (bottom-up) the safety lifecycle (see Section 3.5.1 for details). Moreover, through a realistic example, we showed that the HSDM model can be the basis for hazard analysis and risk assessment of automotive systems. We also provide UML profiles for such analysis (see Section 3.5.2). Finally, to ensure correctness of HSDM, we used external reviews during modeling on two levels. The first review focused on the UML modeling aspects and the second review focused on content regarding functional safety of HSDM. For the second review we asked a member of ISO functional safety working group (ISO/TC 22/SC 32/WG 8) for reviewing the results. We reflect on this review with the following subsection.

3.6.1 External review

The technical review resulted on some technical discussions on two topics regarding details of the HSDM models, and finally a statement from the reviewer. Here, we present a summary and conclusions of the technical discussions.

Item The term “Item” was criticized on multiple accounts: First, it is a nomenclature for an already existing concept, i.e., “system.” Second, it prescribes a system hierarchy throughout the whole document that may not be true for all the systems within the scope of this standard.

While we agree with this comment, we could not rectify the comment because of two points in the definition of Item (as follows). First, an Item is not any system; but it is a system to which ISO 26262 is applied, which we interpret as the development of the system is in compliance with ISO 26262. Second, an Item is a system with functions at the “vehicle level”.

Definition 4. *"Item is a system or combination of systems, to which ISO 26262 is applied, that implements a function or part of a function at the vehicle level."*

While the term "vehicle level" is not defined, it is used for defining multiple important terms in this standard such as Safety goal and Intended functionality. The solution that we consider for this issue in HSDM is to single out Item (to capture the "vehicle level" system) and refer to all the decomposed systems as Element. This solution requires the same construct on the behavioral aspects and considering Item Intended Function separately than other Functions (allocated to Element).

Hazard Consequence The term "Hazard consequence" and the concepts related to it are criticized on the following comments: First, hazard *is* the malfunctioning behavior and is not *caused* by it. Second, "hazardous event" occurs when a hazard is situated in an operational situation, therefore is deemed enough for effective hazard analysis, therefore "hazard consequence" is not needed.

In answer to the comments, we present the definition of Hazard and the relevant clauses about Hazardous Event:

Definition 5. *"Hazard is a potential source of harm caused by malfunctioning behaviour of the item."*

> Part 3 6.4.2.1: "The operational situations and operating modes in which an item's malfunctioning behaviour will result in a hazardous event shall be described; both when the vehicle is correctly used and when it is incorrectly used in a reasonably foreseeable way."
>
> Part 3 6.4.2.6: "The consequences of hazardous events shall be identified."

On the basis of these three statements, we interpret the following: First, Hazard is caused by malfunction of an Item. Second, Operational situations and operating modes relevant for a "Hazardous event" are required. Third, consequences of a Hazardous event is also required. These interpretations, results directly in the solution as explained in Section 3.4.3.

The technical review process resulted in the following statement from the external reviewer:

> "This work proposes a tool based approach (HSDM) to perform a static analysis of system requirements against ISO 26262 /ISO21448 standards. Complex system of systems implemented at any hierarchal level, can now run an excess of tens of thousands of pages. In the future, checks for completeness, coexistence and independence will rely on computer based approaches as suggested in this paper. In a similar way in which compilers now check for compliance against MISRA standards, HSDM will analyze, Model Based System Design and Specifications (MBSE & MBSS) for non-conformances against the stated safety requirements. Hence this work will ultimately increase the confidence that necessary safety assurance for any target system has been achieved."

3.7 Conclusions

In this chapter, we introduced the Holistic Safety Domain Model (HSDM). We based this model on a systematic analysis of the specification of the ISO 26262. The systematic modeling method ensures the alignment of the resulting model and the text of the standard. Our proposed domain model formalizes the specification of the standard and captures both the system design and process aspects. We show two applications of our proposed model: modeling the work flows of the safety lifecycle, and modeling safety analysis specifications. For the later application, we provide a UML profile that allows modeling the concepts discussed in HSDM.

Modeling the standard ensures consistent interpretation of its text and reduces the risk of human errors in communication. Using a model-based approach, we ensure the traceability of the information (one of the essential requirements of the ISO 26262) and a unique interpretation of the analysis and requirements. As for future work, the HSDM and the formalized work flows can be used for automatic checks on compliance by modeling the constraints on the concepts. Another application of HSDM could be (partially) automating some of the safety activities, thus reducing the efforts of safety engineering. Such automation requires sufficient software tool support for effective use.

Chapter 4

Compliance assurance for automotive safety-critical development: a model-based approach

In this chapter, we present a novel approach that supports the development of standard-compliant systems based on model-based techniques. We use a domain model of ISO 26262 that covers both process and system design aspects. We then define constraints that define non-compliance to this standard. We check the constraints before or after the related safety activity. This way, we can discover compliance errors at the moment they happen; and depending on the type of fault, we formulate feedback, or apply an automatic fix and inform the user. To prove the concept of our approach, we developed a software tool. This tool keeps a common project model between safety engineers and system engineers. In our setting, the system engineer uses Enterprise Architect to specify the system functions, and the safety engineer uses MS Excel to perform hazard analysis. The proposed tool executes the constraints of the domain model and provides feedback. The proof of concept includes the first two major safety activities according to the ISO 26262 safety lifecycle: Item definition and Hazard Analysis and Risk Assessment. We show that we can detect compliance errors even in a multidisciplinary project setting. We believe that detecting design faults in the correct moment decreases the chance of human errors to become design errors.

This work is based on:

[67] A. Khabbaz Saberi, D. van den Brand, and M. G. J. van den Brand “Towards compliance assurance for automotive safety-critical development: a model-based approach,” *in the poster session of the 6th International Symposium on Model-Based Safety and Assessment (IMBSA 2019),* 2019

4.1 Introduction

Compliance to norms and standards is an effective tool to achieve safety and system quality. In the automotive domain, the ISO 26262 [52] standard captures the state of the art of functional safety. The standard provides guidelines on two aspects of safety-critical development: the process guidelines and product based requirements. It indicates the work products that are required and produced during all the safety activities in the defined safety lifecycle. Besides, it mandates safety analyses on multiple levels of system development, e.g., the Hazard Analysis and Risk Assessment (HARA) that is required during the conceptual phase.

Moreover, this standard requires evidence of compliance in a safety case to assure that a system is safe. The evidence should provide sound reasoning that creates trust in the safety of the system. The safety case covers the system design safety, i.e., why the final product is safe from a technical perspective, functional safety that covers how the product behaves as intended in case of failure, and the process aspect that ensures sufficient and correct development processes during life cycle of the product. Previous research has studied methods for assessment of the quality of a safety case [84].

Furthermore, the ISO 26262 standard imposes strict requirements on traceability between safety requirements that are resulting from various safety analyses and design elements, which are part of the architecture of the system. The need for more integration of safety engineering processes with system engineering is noticed [85].

These trends increased the complexity of the systems. We can see this complexity from various angles: Firstly, the systems consist of more components. As a result of increased demands on functionalities, more hardware parts and software units are deployed on a vehicle to realize the automated driving or smart mobility features. Secondly, various domains of expertise are required to work together to achieve these features. Expertise such as software engineering is needed to develop a vehicle, next to the more traditional domains such as mechanical engineering.

These aspects of complexity increase the chance of design errors. There are more components in the system. Therefore more design effort is required for system development, and more design errors may occur. Moreover, it is more difficult to detect these design errors since there are more dependencies among components. Another source of design errors may be the miscommunication among a multidisciplinary development team as the individuals may have a different understanding of the system. An indication of these issues can be seen in the number of software-related recalls during recent years. For the major vehicle manufacturers, the number of software related recall campaigns have grown steadily from 2011 to 2015 by nine times [120].

In this research, we address the compliance assurance of safety engineering. We limit the scope of this chapter to the concept development phase to focus on the safety analyses that are performed in an early phase of development in the automotive domain.

In ISO/IEC/IEEE 42010 [49] an "architecture viewpoint" is defined as:

Definition 6. *"work product establishing the conventions for the construction, interpretation, and use of architecture views to frame specific system concerns".*

Considering this definition, we can see safety analyses as different architecture views within the safety viewpoint. The impact of safety requirements on architecture design has been the subject of study in the literature [79].

The idea of using model-driven techniques to support safety engineering is not new. Several publications in the literature address this topic [4], [7], [12], [85], [124]. However, most of the papers focus on the system design aspect of the safety engineer and attempt providing a modeling language for expressing safety concepts. Very few publications mention the importance of the development processes for functional safety assurance [18], [73].

There are already some related work in the literature on the formalizing the ISO 26262 standard. The OPENCOSS project[1] aimed at providing a common certification framework between avionics, railway and automotive concerning safety. This project resulted in a generic meta-model for safety standards, called SafetyMet [129], which provides a framework for modeling a safety standard. Because a common framework is created for different standards, the models are at a high abstraction level. Here, a conceptual model describes concepts and terms from the standard; and the behavioral model shows when, and by who, these concepts must be created during development. Our work makes a clear distinction between the conceptual model, and the behavioral model, and provide a more detailed description of each of these aspects.

In avionics J. Wu *et al.* formalized the DO-178B/C standard [135]. Their work aims to facilitate safety oriented architecture. Similar to our research, they focused on system development. With the help of domain experts, safety properties have been extracted from the DO-178B/C standard. These were then further refined into constraints on a domain model for avionics. Compared to DO-178B/C the ISO 26262 standard is less mature and contains some vague statements. This vagueness leaves some open challenges in how the application of constraints can aid design in the automotive industry.

The contribution of this chapter is providing a pragmatic solution for integrating the system design and process aspects. The novelty in our contribution is that we achieve this integration by applying constraints to the domain model. Also, we provide a proof of concept by showing an example of how the constraints are applied in practice. We use commonly used software tools in the industry(MS Office and Enterprise Architect), and demonstrate the applicability of our approach in detecting non-compliance.

By viewing the hazard analysis as a model, in this chapter, we propose a model-based engineering approach to detect non-compliance and minimize the impact of possible errors. By defining non-compliance regarding model constraints, we automate the detection of these errors. Since the verdict of an automated test is less influenced by the growing complexity of a system; we argue that our approach can benefit the development of complex automotive systems.

The rest of this chapter is organized as follows. In Section 4.2 the proposed methodology is explained in more detail. In Section 4.3 it is shown how we apply this methodology with an example. Finally, in Section 4.4 we conclude our findings.

4.2 Methodology and Results

In this section, we describe our approach for creating a formal notation in the form of a domain model. Furthermore, we explain how the domain model can be used to create a formal specification of compliance with the ISO 26262.

[1] Open Platform for Evolutionary Certification of Safety-critical Systems - `http://www.opencoss-project.eu/`

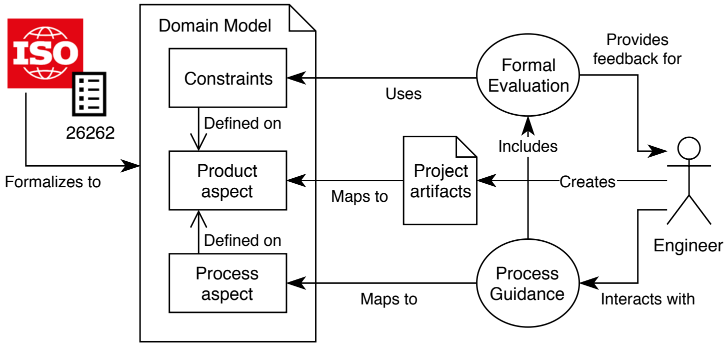


Figure 4.1: Overview of the methodology

4.2.1 Domain Model and Evaluation

As we show in Figure 4.1, we assume that the domain model consists of the system design and process aspect (similar to the model described in Chapter 3). The system design part of the domain model provides a classification for all the concepts required for safety analysis. This aspect classifies the information in the domain model into Classes, Attributes, and Relations between Classes (following a UML construct). The process aspect consists of activities that have input and output from the system design aspect. The inputs and outputs of an activity denote the needed and produced information respectively. The process aspect denotes the development processes that must be executed (i.e., the safety lifecycle). The process aspect also includes the required concepts (form the system design aspect) for each activity. We denote the compliant process execution as process guidance. The compliant process is what *should* happen and what is suggested to the user (hence the name).

During a the development process, information is generated in various forms. We refer to all the information generated during the system development as well as their specifications as *project artifacts*. In the terminology of ISO 26262, the documented collection of these project artifacts are called a work product (typically the output of a major activity). We define constraints that specify what should hold in these project artifacts at certain stages of development to comply with the standard. To achieve this, the constraints are defined regarding the system design aspect of the domain model. In this way, the constraints can evaluate the information present (or missing) in the project artifacts. Each constraint is related to an activity from the process aspect of the domain model. This allocation defines when the constraint must be evaluated, it should hold before the activity is started or after it has finished. By providing the engineer with detailed guidance through the process aspect of the domain model the failure of certain constraints will give useful feedback about the project artifacts.

4.2.2 Defining the Constraints

We inspect the ISO 26262 requirements to identify constraints. The majority the ISO 26262 specifications are guidelines on information that must be included in a safety work product. The ISO 26262 standard is concerned with two aspects: *methods* for determining the information and the *quality* of this information. The first step for defining the constraints is to identify the subject of each constraint. We use the subject to determine the context of the

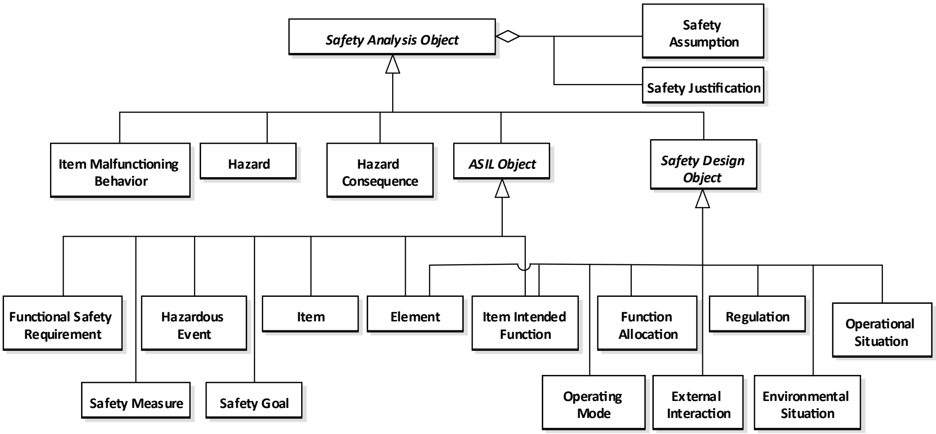


Figure 4.2: System design aspect of the domain model, note that this is the same model as in Chapter 3

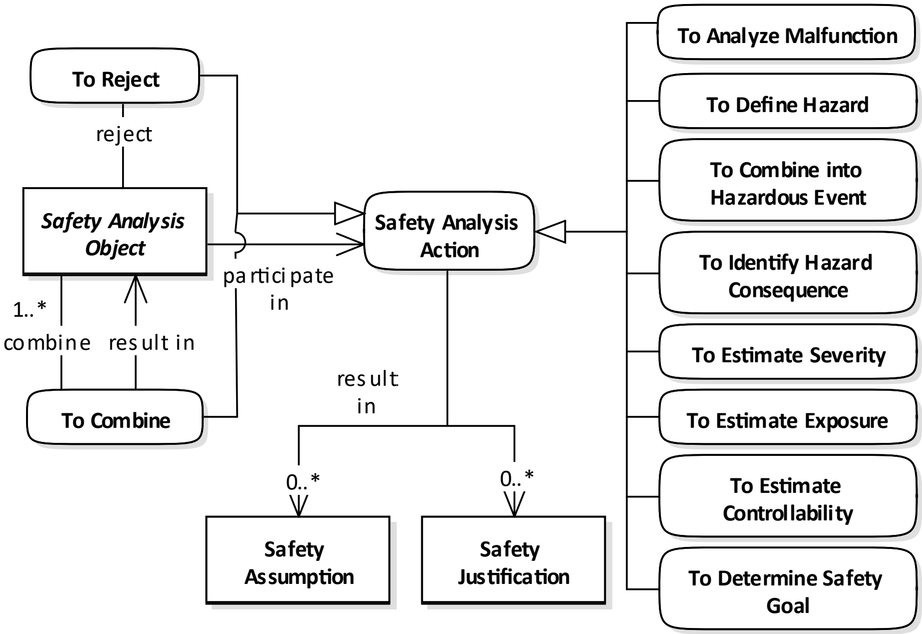


Figure 4.3: Process fragment of the domain model, note that this is the same model as in Chapter 3

constraint. We then write them informally in natural language. We use the informal text for providing feedback when a constraint is violated. We trace the constraints to a clause from the ISO 26262 document, such that the constraint violations can be assessed within the context of the ISO 26262 requirement. Finally, we express the resulting constraints in

Eclipse Validation Language (EVL[2]).

We formulate the *quality* aspect requirements in terms of constraints too. The quality requirements are either subjective or objective. The requirements related to ASIL are good examples of the objective requirements. It is not possible to fully formalize the subjective quality requirements. However, some indications as to whether such requirement is satisfied can be included. For example, whether enough information is available in the *item definition* work product cannot be formalized. In these cases, expert's input is required, and constraints simply check the input.

We assign the constraints from the ISO 26262 standard to the domain model via different approaches. The first approach consists of an inspection of the system design aspect of the domain model. Each relation in the system design aspect can have a certain cardinality associated with it. By inspecting every relation separately and searching through the related requirements in the ISO 26262, we identify these constraints. An example is shown in Figure 4.2, where the hazard analysis part of HARA is shown. An inspection of the causes relation and related requirement indicates that it would be invalid to have a Hazard which is not caused by an Item Malfunctioning Behavior.

Another type of constraints come from the Attributes. Although the ISO 26262 standard does not always explicitly define that each element should have a name and description it would be problematic if these are not defined.

Finally, we use the process aspect for assigning constraints. We associate constraints with a relation between a class and an activity from the process models. For example, in the process aspect of the HARA in Figure 4.3, the activity To Determine Hazard has one input and two outputs. The input is a Item Malfunctioning Behavior; and the outputs are a Hazard (the primary output) and (justified by) a Safety Justification. The constraint that should hold among these relations is that a there should be a Safety Justification that justifies a Hazard resulting from an Item Malfunctioning Behavior.

Here we give some examples of the identified constraints. For instance, Clause Part 3–6.4.4.1 and 6.4.4.2 specify:

> The operational situations and operating modes in which an item's malfunctioning behavior will result in a hazardous event shall be described; both when the vehicle is correctly used and when it is incorrectly used in a reasonably foreseeable way.

and

> The hazards shall be determined systematically based on possible malfunctioning behavior of the item.

```
// Constraint 1:
context HazardousEvent {
  message: 'Every hazardous event should be justified'
  // JustifiedBy refers to the relation between
  // Safety Analysis Object and
  // Safety Justification in Figure 4.2
  inv: JustifiedBy.exists(jb | jb.safety_analysis_object=self
    and jb.safety_jusitifcation)
}
```

[2] Eclipse Validation Language - https://www.eclipse.org/epsilon/doc/evl/

```
// Constraint 2:
context Hazard {
  message: ’The combination of every Hazard, Operating Mode
            and Operational Situation should lead to a
            Hazardous Event, or be justified’
  inv: OperatingMode.all(om |
    OperationalSituation.exists(os |
      // ToCombineIntoHazardousEvent refers to the action
      // in Figure 4.2
      ToCombineIntoHazardousEvent.exists(a |
        ParticipateIn.exists(p | p.safety_analysis_action=a
          and p.safety_analysis_object==self) and
        ParticipateIn.exists(p | p.safety_analysis_action=a
          and p.safety_analysis_object==om) and
        ParticipateIn.exists(p | p.safety_analysis_action=a
          and p.safety_analysis_object==os) and
        (
          // Either there is a Hazardous Event, that is
          // connected to each Hazard, Operating Mode and
          // Operational Situation (as seen in Figure 3.6.a)
          HazardousEvent.exists(he |
            ParticipateIn.exists(p | p.safety_analysis_action=a
              and p.safety_analysis_object==he) and
            (
              AggregationOf.exists(r | r.hazardous_event=he
                and r.hazard=self) and
              InCombinationWith.exists(r | r.hazardous_event=he
                and r.operating_mode=om) and
              OccursIn.exists(r | r.hazardous_event=he and
                r.operational_situation=os)
            )
          )
          or
          // Or, such an hazardous event does not exists and
          // the action must give a Safety Justification
          // (as seen in Figure 3.4.b)
          SafetyJustification.exists(sj |
            ResultIn.exists(p | p.safety_analysis_action=a and
              p.safety_justification==sj)
          )
          // Note: This justification can not be linked to a
          // Hazardous Event object.
        )
      )
      Causes.exists(c |
        c.h==self and c.om=om and c.os=os and (c.he or c.j)
      )
    ))}
```

Table 4.1: Specified constraints from the ISO 26262 standard

ISO part	ISO requirement	Number of constraints	Name	Category
3	8.4.1	1	Functional safety requirements is a safety requirement	M
	8.4.2.1	1	Functional safety requirements are derived from safety goal	M
	8.4.2.2	1	Each safety goal leads to al least one functional safety requirement	M
	8.4.2.4	2	State transitions to a safe state need to have a condition	M
	8.4.2.4	1	There must be a transition for safe states	W
	8.4.2.5	1	Safe state needs emergency operation	M
	8.4.3.2	3	Functional safety requirements are based on 1 element of the same item	M
		6	Information traverses the allocation relation of FSR	M
		1	Functional safety requirements derived from others, in case of systems	M
	8.4.3.3	1	External technologies need interface requirements	M
	8.4.3.4	2	External risk reduction shall be ensured	M
	8.4.6	1	Safety validation criteria should be defined	M
8	6.4.1.1	3	Informal, formal or semi-formal notation for certain ASIL	S
	6.4.2.2	1	Safety requirements inherit ASIL	M
	6.4.2.3	1	Safety requirements are allocated to item or element	M
	6.4.2.4	6	Safety requirements are validated to be: unambiguous, comprehensible, atomic, internally consistent, feasible and verifiable	M
	6.4.2.5	1	Requirement ID is unique	M
	6.4.3.1	1	Safety requirements cannot have lower ASIL then their parents	M
	6.4.3.2	2	Safety requirements are traceable with respect to: parents and elements	M

In this example the Causes class has five attributes: h (Hazard), he (Hazardous Event), om (Operating Mode), os (Operational Situation), and j (Justification). These constraints are both associated with To Identify Hazardous Event and are to be evaluated after the execution of this activity.

4.2.3 Results

We have selected two chapters from the ISO 26262 standard and created the corresponding constraints: Section 8 of Part 3 and Section 6 of Part 8.

For every requirement we have identified some constraints. Some examples can be seen in Table 4.1. In total, we identified 53 constraints from these sections. We give a short descriptive name to each constraint to indicate the content of each constraint. Also, we categorize each constraint in three categories: Must Satisfy (M), Suggested (S), and Warning (W). The category denotes the severity of a violation of the constraint. M (must satisfy) means the constraint must hold to comply with the standard. S (suggested) means that it is merely and advice from the standard. And W (warning) denotes elements which we identified as problematic when trying to comply to the standard. This last category contains constraints which are normally inferred by the reader of the text, however, it is technically not needed in order to comply with the standard.

4.3 Proof of Concept

To prove the applicability of our proposed approach, we developed a software tool. In this section, we describe the technical design of the tool. Then we demonstrate how the tool is applied for safety analysis.

4.3.1 The Tool Context

The tool is developed for a project with both safety and system engineers to reflect on the multidisciplinary nature of automotive systems, as seen in Figure 4.4. Engineers use different tools during development; we consider Microsoft Excel for safety engineers and Enterprise Architect (EA) for system engineers. For a specific project the domain model is instantiated that is called the project model. Each engineer interacts with the project model through a plug-in in their already known development tool. This plug-in is developed to provide the required functionalities of our tool to the engineer.

To demonstrate different ways of working, our tool keeps a project model between two different kind of tools: document and model-based tools. More specifically, the safety engineers work in a document-based tool like Excel and the system engineers work in a model-based tool like EA.

4.3.2 Tool functionalities

In this section, the following four major functionalities of our tool are described.

(1) Maintain a *common project model* between the different tooling and users. In the model-based tool the mapping to the common project model can be automatically extracted by means of a one-to-one mapping from the meta-model of the tool to the domain model. However, in Excel this mapping has the be defined based on the columns. To support the use of different templates in Excel the user is allowed to specify this mapping in the Excel plug-in. In conclusion, the mapping from the domain model to EA and Excel will enable the tool to automatically keep the project artifacts and the system design aspect of the project model synchronized.

(2) Provide process *guidance* to the user. The tool enables the user to specify which steps in the domain model are to be taken. Via the user interface in EA and Excel the user can initiate the actions and advance through the process aspect of the domain model.

(3) *Execute constraints* on the project model. For every action, the pre- and post-constraints need to be executed. Because the domain model is filled in from different clients

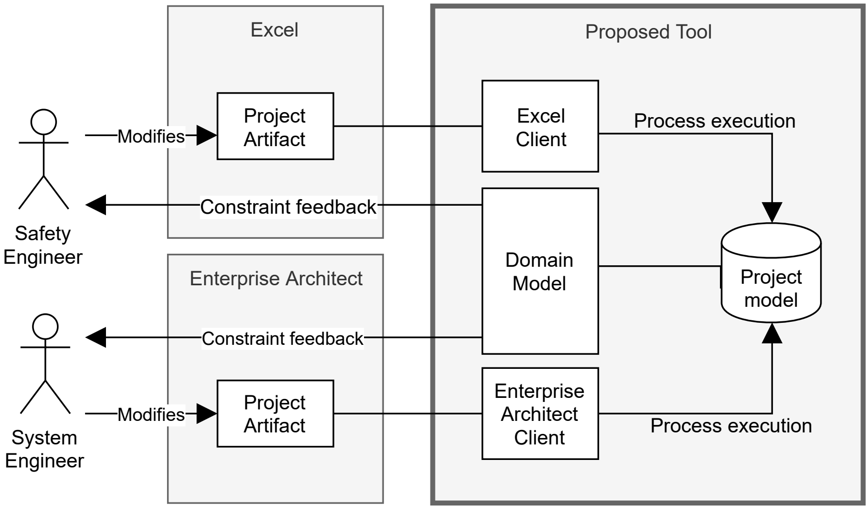


Figure 4.4: Tool context overview

these constraints should be executed in a central location. Based on the most common constructs that were found a custom language was created to specify these constraints. In certain cases, the ISO 26262 standard is very precise and an automatic fix for the constraint is possible to be specified.

(4) Give *feedback* to the user on the evaluation of these constraints. Constraints are evaluated on the project model, but the actual initiation of this evaluation happens in the client. So the feedback should be integrated into the client and shown in the project artifact. Also, useful information should be displayed which helps the user to fix the mistake.

4.3.3 Tool usage example

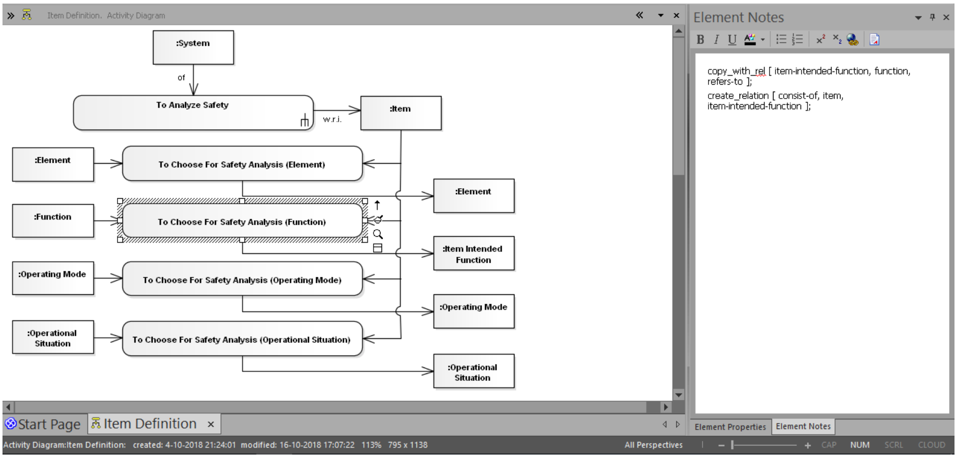
This section explains how the tool was initially configured. And then it is explained how the tool was used during the Hazard and Risk Analysis (HARA).

4.3.3.1 Initial domain model setup

To validate the approach a part of the domain model with constraints needs to be specified in the tool first. For this, we specify a part of the domain model, as shown in Figure 4.2 and 4.3. In addition, we specify a part of the process as well, as shown in Figure 4.5.

4.3.3.2 The HARA process

After the preliminary elements of the system architecture are designed the system engineer must decide which elements can be used for safety analysis. For these elements, the work is handed over to the safety engineer. The safety engineer will perform the safety analysis and will need to identify the malfunctioning behavior that an ACC system can express. To achieve this, a HAZard and OPerability (HAZOP) analysis is performed. In this technique, each functionality is compared to HAZOP keywords to identify what kind of malfunction-



Item Definition. Activity Diagram
:System
of
To Analyze Safety
w.r.t.
:Item
:Element
To Choose For Safety Analysis (Element)
:Element
:Function
To Choose For Safety Analysis (Function)
:Item Intended Function
:Operating Mode
To Choose For Safety Analysis (Operating Mode)
:Operating Mode
:Operational Situation
To Choose For Safety Analysis (Operational Situation)
:Operational Situation
Start Page
Item Definition
Element Notes
Element Properties
Element Notes
Activity Diagram:Item Definition: created: 4-10-2018 21:24:01 modified: 16-10-2018 17:07:22 113% 795 x 1138
All Perspectives
CAP
NUM
SCRL
CLOUD

```
copy_with_rel [ item-intended-function, function,
refers-to ];
create_relation [ consist-of, item,
item-intended-function ];
```

Figure 4.5: Process fragment of the Hazard Analysis and Risk Assessment in the domain model (in EA), the constraint for the selected action is shown in the Notes window on the right.

ing behavior can be expected. Each HAZOP keyword and function should result into a malfunctioning behavior or a justification why it would not.

4.3.3.3 Tool usage during HARA

For the tool this implies that elements need to be transfered from the design domain to the safety domain. It is important that changes to these elements in the design domain are not directly affecting the changes in the safety domain during the safety analysis. Therefore in the domain model, this step is modeled as an action which copies the elements from the design domain to their respective counterparts in the safety domain. As can be seen in Figure 4.5 on the left hand there are the design classes (System, Element, Function, Operating Mode and Operational Situation), and on the right there are the safety classes (Item, Element, Item Intended Function, Operating Mode and Operational Situation). The ISO 26262 specifies in Requirement 7.3.1 of Part 3 [52] that during the HARA the elements from the Item definition are considered. It is interpreted that the project cannot be ISO 26262 compliant if the following two constraints do not hold when the HARA activity starts:

1. The elements from the safety domain must refer to a class from the domain model.

2. The elements in the safety domain should be part of some Item element.

More specifically, these two statements should hold when the system engineer executes the "To choose for safety analysis"-action. This action is specified in the domain model, as seen in Figure 4.5 The operation of copying these objects from the design domain to the safety domain can be automatically executed. This is described by translating the two constraints to a custom syntax used by the tool, namely:

1. copy_with_relation[item-intended-function, function, refers-to];

2. create_relation[consists-of, item, item-intended-function];

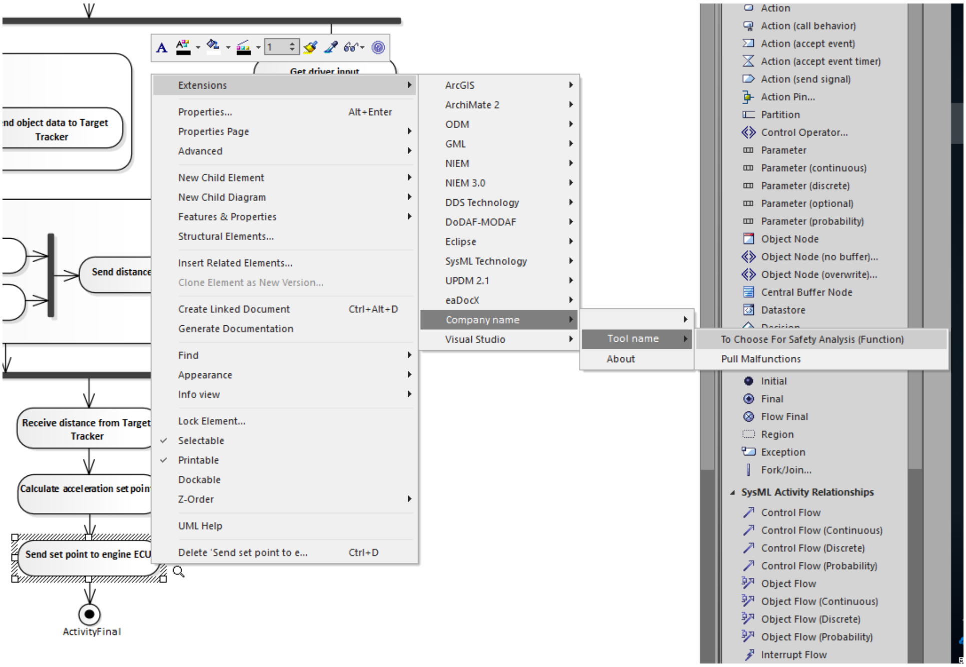


Figure 4.6: EA creation of an Item Intended Function

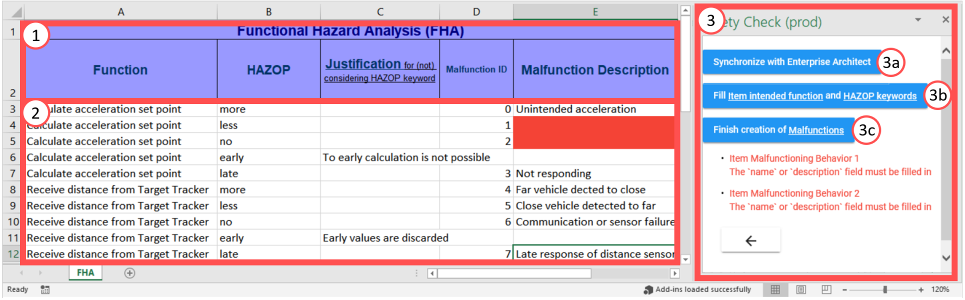
The system engineer uses the context menu of the ACC system in EA to create the item element first. Then for each function, the “To choose for safety analysis (Function)” actions are executed as well, see Figure 4.6. This action will additionally pop up a dialog which makes you select the Item for which the Item Intended Function belongs to.

After this action, the safety engineer resumes work in Excel. A screenshot of the Excel plugin is shown in Figure 4.7. The safety engineer starts with a blank template, where only region 1 of Figure 4.7 is filled in. Using the plugin panel (region 3 of Figure 4.7) the mappings from the columns to the classes from the domain model are created, as shown in Table 4.2. In addition the relations are mapped, in this case only the causes relations from the domain model is presented in this template between columns A, B, C, and D.

Table 4.2: Excel template column to class mapping

Column	Class	Attribute	Parent column
A	Item Intended Function	name	
B	HAZOP keyword	name	
C	Justification	description	
D	Item Malfunctioning Behavior	id	
E	Item Malfunctioning Behavior	name	D

After the mapping for the template is set up, the safety architect presses the “Fill intended function and HAZOP keywords” button to fill in the sheet according to the project


1
2
3
3a
3b
3c
ety Check (prod)
Synchronize with Enterprise Architect
Fill Item intended function and HAZOP keywords
Finish creation of Malfunctions
←
FHA
Ready
Add-ins loaded successfully
120%

Functional Hazard Analysis (FHA)

	A	B	C	D	E
2	Function	HAZOP	Justification for (not) considering HAZOP keyword	Malfunction ID	Malfunction Description
3	ulate acceleration set point	more		0	Unintended acceleration
4	culate acceleration set point	less		1	🟥
5	Calculate acceleration set point	no		2	🟥
6	Calculate acceleration set point	early	To early calculation is not possible		
7	Calculate acceleration set point	late		3	Not responding
8	Receive distance from Target Tracker	more		4	Far vehicle dected to close
9	Receive distance from Target Tracker	less		5	Close vehicle detected to far
10	Receive distance from Target Tracker	no		6	Communication or sensor failure
11	Receive distance from Target Tracker	early	Early values are discarded		
12	Receive distance from Target Tracker	late		7	Late response of distance sensor

- Item Malfunctioning Behavior 1
  The `name` or `description` field must be filled in
- Item Malfunctioning Behavior 2
  The `name` or `description` field must be filled in

Figure 4.7: Excel plugin of our tool detects failure of a constraint and gives feedback

model (button 3b in Figure 4.7). This will automatically fill in column A and B from region 2 with the data from the EA client. Then the safety engineer executes the safety analysis and fills in columns C, D, and E in region 2. For each HAZOP either a malfunction or a justification must be specified.

4.3.3.4 Tooling feedback to the user

After the safety engineer finishes the identification of malfunctions the “Finish creation of Malfunctions” button is pressed (button 3c in Figure 4.7). This finished the execution of the action and triggers the post constraint to be evaluated. In this case, the constraint checks if each HAZOP and Function combination has a Justification or Malfunction with a description. If two objects are not valid according to the constraint, the corresponding cells are given a red background. This shows the engineer that a specific constraint has failed for these two Item malfunctioning behavior objects in the project model. In the panel (region 3) a description of the constraint is shown which hints to why the constraint has failed. Via this mechanism, the engineer will notice clearly that he made a mistake and he or she is shown where this mistake is made. The engineer then takes a closer look at the malfunction objects 1 and 2 and fixes his or her mistake. When the button is pressed again, the cells turn transparent and the error messages disappears. Now the engineer knows that this step is executed and all constraints related to this step are valid.

4.4 Conclusions

An increase in system complexity increases the chance of human error during design and the effort required for quality assurance. In this chapter, we proposed the use of constraints to detect noncompliance during safety engineering according to ISO 26262.

We define constraints based on ISO 26262 and a domain model of this standard. We also provide a software tool to prove the concepts of our proposed method in a multidisciplinary project environment. We designed and implemented this tool to interface with two other software tools, i.e. Microsoft Excel and Enterprise Architect (EA), and to keep a common project model. Our tool evaluates the defined constraints on the project model. We demonstrated that the tool transfers information between Excel and EA. Our tool automatically checks the constraints for failure of compliance to the ISO 26262. Based on this evaluation the tool provides feedback to show certain design mistakes are made. This automation allows checks to be preformed during development of safety work-products and therefore reduce the risk of rework in case of an error. Our proposed method makes it possible to provide feedback in a shorter time period as opposed to checking the models at the end of the development.

Chapter 5

An Architecture Pattern for Safety Critical Automated Driving Applications: Design and Analysis

Introduction of automated driving increases the complexity of automotive systems. As a result, architectural design becomes a major concern for ensuring non-functional requirements such as safety and modifiability. The ISO 26262 standard recommends using architecture patterns for system development. However, the existing architecture patterns may not be able to answer requirements of automated driving completely. When applying these patterns in the automated driving context, modification and analysis of these patterns are needed. In this chapter, we present a novel architecture pattern for safety-related automated driving functions. Additionally, we propose a generic approach to compare our pattern with existing ones. The comparison results can be used as a basis for project specific architectural decisions. Our proposed Safety Channel pattern is validated by its implementation for a real-life truck platooning application.

This chapter is based on:

[79] Y. Luo, A. Khabbaz Saberi, T. Bijlsma, J. J. Lukkien and M. G. J. van den Brand, "An architecture pattern for safety critical automated driving applications: design and analysis," *11th Annual IEEE International Systems Conference (SysCon 2017)*, 24-27 April 2017, Montreal, Quebec, Canada. p. 261-267

5.1 Introduction

Autonomous driving has received significant attention in recent years. It represents the final step in five levels of automation, ranging from no automation (Level 0) to full automation (Level 5) [113]. One of the main motivations for autonomous driving is safety; the argument is that by removing human errors it is possible to achieve higher levels of safety since human error is the most important contributor to fatal accidents. Provided that automated driving systems operate more safely than humans, reaching automation Level 5 results in safer transportation. Realization of higher levels of automation (Level 4, or Level 5) will increase complexity, not only because automated driving applications will need to support more functionality, but also since level-4 and higher applications can no longer rely on human drivers as a fallback of the system. With this, automated driving functions become, even more, safety critical.

The standard for addressing functional safety in automotive is ISO 26262 [52]. This standard provides guidelines for the process of developing safety-related Electrical and Electronic (E/E) systems in a passenger car. Adherence to this standard has been the center of attention in many organizations in recent years, and different approaches have been proposed for applying it [11], [42], [66]. Safety is rooted in the system architecture, as decisions about the architecture of a system have a great impact on (or better: determine) qualities of that system like performance, reliability, and safety. One of the recommendations of ISO 26262 is to use well-trusted architecture principles, which are traditionally expressed as architecture patterns or styles. The impact of architecture patterns is also addressed in other literature [13], [87].The ISO/IEC/IEEE 42010 standard [57] recognizes architecture patterns as a fundamental way for classifying designs and retaining experience. Besides, adherence to architecture patterns is considered a form of redundancy [127].

Architecture patterns that address safety should enhance fail-safe[1] and fail-operational[2] properties while simplifying and standardizing the design process. In this chapter, we propose a novel architecture pattern, suitable for automated driving functions. The pattern is built around functional safety and is applicable in situations with conflicting safety goals. It includes typical constraints found in the automotive industry like embedding, real-time execution, and implementation costs. The second contribution of this chapter is a generic approach to comparing different safety architecture patterns concerning many quality attributes derived from the quality model in the standard ISO/IEC 25010 [51]. The comparison can be used as a basis for project specific architectural decisions. We validated the pattern by implementing it in a truck platooning application, which aims for Level 3 public road demonstration.

The remainder of this chapter is organized as follows: Some background is presented in Section 5.2. We introduce and describe the details of our proposed architecture pattern in Section 5.3. In Section 5.4, we present a systematic comparison of some of the safety patterns. The description of validation of the proposed pattern is given in Section 5.5. Finally, Section 5.6 concludes this chapter.

[1] Remain safe under a given failure model.

[2] Remain operational under a given failure model.

5.2 Background Information

In this section, we present background and related work on functional safety, and safety architecture patterns.

5.2.1 ISO 26262

The ISO 26262 standard is an adaptation of IEC 61508 [47]. It provides a framework for developing safety-related systems in the automotive domain. It recommends various methods for designing, analyzing, and testing safety-related systems. One of the safety analyses required in an early development phase is Hazard Analysis and Risk Assessment (HARA). Here, we give a short introduction to HARA and ASIL decomposition. We use these safety activities for the discussion on the safety aspects of our proposed architecture pattern.

HARA is one of the first steps in the concept phase of ISO 26262. It provides a procedure for identifying hazards, classifying their risk using Automotive Safety Integrity Level (ASIL) rankings, and defining safety goals aiming to prevent or mitigate the identified hazards. ASIL is a mechanism for classifying the risk associated with a hazard. The ASIL ranking ranges from ASIL A, indicating the lowest risk class to ASIL D, the highest risk class. The ASIL rankings are assigned to the hazards and inherited by the defined safety goals. After deriving or refining safety requirements from safety goals, the ASIL ranking is also inherited by the requirements[3]. Subsequently, after allocation of the safety requirements to architectural elements, the development requirements of those architectural elements are set in accordance with the highest ASIL assigned to them.

ASIL decomposition is a design method recommended by ISO 26262. This method allows designers to share the safety responsibility between independent architectural elements. In this method, a requirement with a higher ASIL can be refined into two (or more) redundant requirements with lower ASIL rankings. The refined requirements should be allocated to independent architectural elements. Table 5.1 shows an example of possible ASIL decomposition. Note that ASIL decompositions have to be backed up with sufficient evidence proving the independence of the involved architectural elements. Here, independence means the absence of any shared single point failure among the elements.

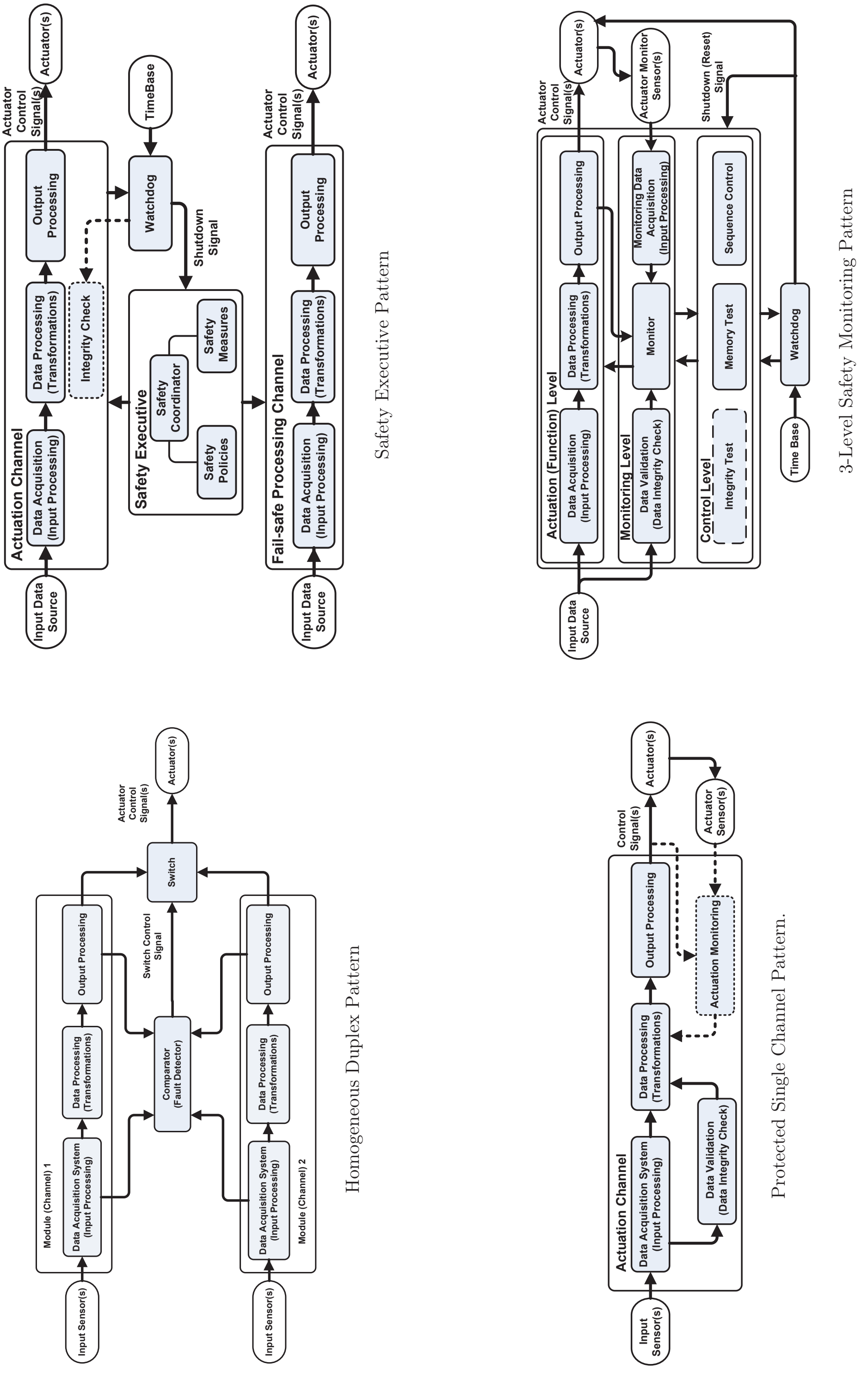
5.2.2 Architecture Patterns

An architecture pattern expresses fundamental decisions governing the design of a system. It offers solutions to accommodate common concerns of a system, typically the non-

Table 5.1: ASIL decomposition example: The possibilities for decomposing an ASIL D requirement (based on requirements of Part 10 of ISO 26262)

Original ASIL	Possible decomposition arrangements
ASIL D	ASIL C(D) and ASIL A(D)
	ASIL B(D) and ASIL B(D)
	ASIL D(D) and ASIL QM(D)

[3] In case a requirement is derived from several goals, then the highest ASIL ranking is inherited by that requirement.


Input Data Source
Actuation Channel
Data Acquisition (Input Processing)
Data Processing (Transformations)
Integrity Check
Output Processing
Actuator Control Signal(s)
Actuator(s)
TimeBase
Watchdog
Shutdown Signal
Safety Executive
Safety Policies
Safety Coordinator
Safety Measures
Fail-safe Processing Channel
Input Data Source
Data Acquisition (Input Processing)
Data Processing (Transformations)
Output Processing
Actuator Control Signal(s)
Actuator(s)
Input Data Source
Actuation (Function) Level
Data Acquisition (Input Processing)
Data Processing (Transformations)
Output Processing
Actuator Control Signal(s)
Actuator(s)
Actuator Monitor Sensor(s)
Monitoring Level
Data Validation (Data Integrity Check)
Monitor
Monitoring Data Acquisition (Input Processing)
Shutdown (Reset) Signal
Control Level
Integrity Test
Memory Test
Sequence Control
Watchdog
Time Base
Input Sensor(s)
Module (Channel) 1
Data Acquisition System (Input Processing)
Data Processing (Transformations)
Output Processing
Comparator (Fault Detector)
Switch Control Signal
Switch
Actuator Control Signal(s)
Actuator(s)
Input Sensor(s)
Data Acquisition System (Input Processing)
Module (Channel) 2
Data Processing (Transformations)
Output Processing
Input Sensor(s)
Actuation Channel
Data Acquisition System (Input Processing)
Data Validation (Data Integrity Check)
Data Processing (Transformations)
Actuation Monitoring
Output Processing
Control Signal(s)
Actuator(s)
Actuator Sensor(s)

Safety Executive Pattern

3-Level Safety Monitoring Pattern

Homogeneous Duplex Pattern

Protected Single Channel Pattern.

Figure 5.1: An overview of some of the safety architecture patterns [1]

functional concerns or qualities. An interesting way to study architecture patterns is by considering the most important system quality that they address [87]. In this chapter, we focus on safety as the main quality aspect. There are various architecture patterns for safety-related systems in the literature [1], [22], [47]. To name a few: Protected Single Channel, Homogeneous Redundancy, Heterogeneous Redundancy, Safety Executive, and 3-level Monitoring, where 3-level Monitoring is more commonly known as the E-Gas [3] pattern. An overview of the functional view of these patterns is shown in Figure 5.1. Here we give a short description of these patterns.

All patterns mentioned above are variants of the Channel pattern. A *channel* is a path via which data flows, from its source to its destination; in automotive, this is usually from sensors towards actuators. Figure 5.2 shows the basic elements of a channel. In this view, the inputs include sensors and reference signals from other systems. The input processing function is responsible for converting the input data into useful information. The data processing function analyzes the information calculates the control signals for the actuators. The output processing function translates the control signals, generated by data transformation, for the actuators.

The Protected Single Channel pattern improves safety by monitoring the input data, checking the data integrity, and optionally monitoring the outputs. The data integrity check function verifies the signals received from the sensors. Based on the validity of the information, the data transformation may decide to switch to a safe operating mode.

The Homogeneous Redundancy pattern improves safety and reliability by copying the main channel and switching between the two copies in case of a failure in one of the channels. The Duplex, triple modular, and similar patterns are variations of this one. The Heterogeneous Redundancy pattern has similar logic to Homogeneous Redundancy with the difference that each added channel is developed independently, therefore making it one of the most costly patterns.

The Safety Executive pattern can switch to a secondary channel to bring the system to a safe state in case of a failure in the main channel. This pattern is useful when shutting down the system requires complex procedures.

The 3-level Monitoring pattern is widely used in the automotive industry because it provides a cost-effective safety solution. This pattern monitors the internal states of a system in the first level, monitors the inputs, and outputs in the second level. The third level is responsible for the nominal functionality of the system.

The discussed patterns are primarily aimed to be applied at the System level. However, some of the patterns apply to the Hardware and Software levels. For example, we could build a heterogeneous redundant hardware platform running homogeneous redundant software. If the software detects a failure, it can decide to switch to a secondary channel running on the same hardware platform.

An analysis on the impact of the discussed safety patterns on cost, reliability, safety, negotiability, and execution time has been provided in [1]. Moreover, a template to describe architecture patterns is suggested by [1]. This template suggests to describe the following elements: pattern name, abstract, context, problem, structure, implication, implementation, consequences, and related patterns.

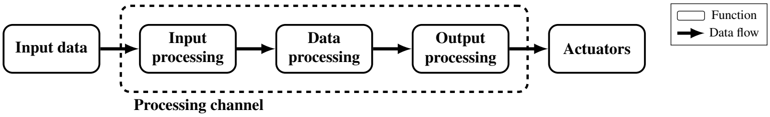


Figure 5.2: The basic channel functional view

5.3 The Safety Channel Architecture Pattern

In this section, we propose a novel architecture pattern, Safety Channel. We follow the design pattern template of [1] to describe Safety Channel. We start with the summary or abstract of this pattern.

The Safety Channel pattern is specially designed considering safety-related highly automated applications in the automotive domain. The goal of the Safety Channel pattern is to provide a strategy where safety is guaranteed even in the presence of severe errors in the nominal functionality. The idea is to reduce the risk of failure of the nominal functionality, implemented by the Actuator channel, by applying ASIL decomposition. Furthermore, the safety responsibility is shifted to a secondary channel (referred to as the Health channel). Inspired by patterns such as Safety Executive [22] and 3-level Monitoring, the Safety Channel pattern aims at simplifying the safety design task, while preserving the reliability and safety of the overall system.

5.3.1 Context

The Safety Channel pattern helps the development of safety-related Automated Driving applications for the Automation Level 3 and higher. These applications have a high fail-safe or fail-operational requirements because switching off the feature does not lead to a safe state. More complex safety measures, such as the gradual degradation of functionality are required to guarantee safety in case of failure.

5.3.2 Problem

The problem addressed by the Safety Channel pattern can be formulated as follows: *How to ensure the safety of complex functions with complex fail-safe strategies in the presence of conflicting safety measures in a cost-effective way?*

5.3.3 Structure

The Safety Channel pattern is based on the Safety Executive pattern [22] and inspired by the 3-level Monitoring (E-Gas) [3]. The schema in Figure 5.3 shows the functional view of this pattern. The Actuator Channel is responsible for delivering the nominal functionality. The Health Channel is responsible for monitoring the Actuator Channel and activating the Limp Home Channel in case of severe failure in the Actuator Channel. The Limp Home Channel provides the emergency operation upon request of the Health Channel.

Most of the elements in the Actuator Channel are same as the Basic Channel described in Section 5.2. Only the Data Integrity Checker is introduced to validate the signals received from the Sensors. The Health Monitor function monitors both the Actuator Channel and the Limp home channel to cross-check the operating modes of different functions. Also, the

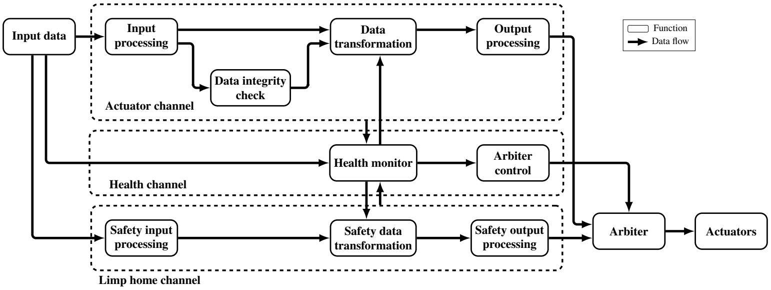


Figure 5.3: The functional view of the Safety Channel pattern.

Health Monitor can detect failures resulted from random hardware faults in the Actuator Channel. The Health Monitor decides on the required degraded mode of functionality and would trigger either the Data Transformation in the Actuator Channel or the Safety Data Transformation to perform the degraded mode.

The Limp Home Channel is similar to a Basic Channel with the difference that the functionality is simplified to the minimum required for bringing the system to a safe state. The Arbiter acts as the switch that is controlled by the Arbiter Control and arbitrates towards the actuators, using signals from the Actuator Channel and Health Channel.

This pattern should be seen in continuation of the similar patterns such as the 3-level Monitoring (E-Gas). The 3-level Monitoring provides integrity by some simple checks on the system and the only available safety measure is restart via watchdog. In our proposed pattern, the concept of health monitor assumes some level of integrity (provided by checking the input data in the nominal channel) and checks on the behavioral properties of the system on a higher level (in comparison with the 3-level Monitoring pattern). Another difference is that the 3-level Monitoring is typically implemented close to the actuator and therefore is on the same channel, while the health monitor can be implemented in a different channel enabling distributed implementation of the health monitoring.

During the functional design phase, the Safety Channel pattern applies to the design of a single AD application or multiple ones. If we design multiple applications using Safety Channel, we may choose to add additional Nominal Channels that are each responsible for a major feature. We advise this strategy when the physical architecture of the additional applications is also separated.

During the technical design phase, each channel is mapped to a separate architecture element (either hardware or software elements). This mapping is required to keep the channel independent (no shared single point fault). In the case of designing multiple features, they may share the Health Channel and Limp Home Channel. Therefore, we reduce the required hardware element that leads to a better cost-effective design.

5.3.4 Implication on Safety

To demonstrate the safety implications of this pattern, we assume that we have an ASIL D requirement for the function of the Actuator Channel. It is possible to decompose this requirement in three refined requirements, each allocated to one of the channels. First, the requirement of Actuator Channel can get a lower ASIL such as ASIL A(D). Second, a new requirement for the transition to the safe state is allocated to the Limp Home Channel with a

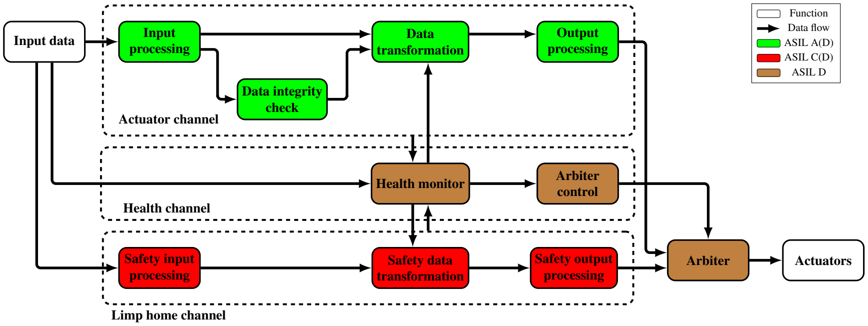


Figure 5.4: The ASIL analysis on Safety Channel.

decomposed ASIL C(D); and third, the Health Channel gets an ASIL D(D) requirement for monitoring and detecting the failure modes of the Actuator Channel. We show the resulting ASIL allocation after ASIL decomposition in Figure 5.4.

5.3.5 Implementation

The most important consideration during the implementation of this pattern is to ensure the independence of each channel. Here, independence means that there should not be a common failure (shared single point of fault), or shared cascading failure between the channels. Moreover, the Limp Home functionality should be more reliable and bring the system to a safe state within the fault tolerant time interval.

5.3.6 Consequences

The Safety Channel pattern offers a modular separation of concerns. Therefore, adding new functionality is possible by adding a new actuator channel. The Health Channel needs to be updated with the failure model of the added channel, effectiveness of Limp Home Channel needs to be verified to accommodate that change. Moreover, extending the functionality implemented in the actuator channel would be possible without any major changes in the other channels, provided that the failure model and interfaces have not been changed.

The drawback of this pattern is the complex implementation of Health monitor. Therefore, this pattern is only useful if achieving the required functionality is relatively more complex.

5.3.7 Related Patterns

As mentioned, the Safety Executive and 3-level Monitoring patterns are closely related to our Safety Channel pattern. The difference between Safety Channel and Safety Executive is that the Health Monitor also uses the data received from inputs to decide on the status of the Actuator Channel. Therefore, it is useful for where the safety of the intended functionality depends on sensing parameters outside of the system. The difference between Safety Channel and 3-level Monitoring is that in Safety Channel the monitoring functionality is allocated to a separate channel and the Limp Home Channel provides basic emergency functions in case of a severe failure in the Actuator Channel.

5.4 Comparison of Architecture Patterns

In this section, we compare our proposed pattern with some existing patterns, which we obtain from the preliminary literature study. The results of the comparison are useful as a reference for making project-specific architecture decisions. The comparison of these patterns is carried out according to five quality attributes: reliability, safety, cost, modifiability, and impact on executive time. We derived these quality attributes from sub-characteristics or attributes of the quality model in the standard ISO/IEC 25010 [51]. For example, cost and impact on executive time are the derivative of performance efficiency, and modifiability is the derivative of maintainability. However, ISO/IEC 25010 does not consider safety as a quality attribute, which we added to support the safety-critical aspect of automated driving applications.

5.4.1 Quantifier of Quality Attributes

For each of the quality attributes, we have five some quantifiers to facilitate the comparison process. For a specific project, architecture pattern decisions can be made by experts using the Pugh method [105]. The Pugh method provides a technique for ranking the multi-dimensional options of an option set. However, we want to avoid comparing different options with the same metric. Because some of the options are not comparable to each other, for example, we cannot compare safety with reliability. Our goal is to make the comparison results more general (independent of the implementation context of a pattern). For specific cases, these results are applicable as a guideline for architectural decision making.

Here, we give a short description of the quantifiers of the quality attributes. We use a combination of a letter and number (ranking the quantifiers) of each quantifier level. A higher quantifier ranking of quality indicates better suitability of the pattern concerning that quality. Note that the comparisons are carried out between a basic system and a system developed using the pattern. An overview of the quantifiers and their descriptions is in Table 5.2.

Reliability This aspect shows the relative improvement in the system's reliability achieved by applying a pattern. The reliability of a pattern is assigned to one of the reliability quantifiers R1, R2, and R3.

Safety This aspect indicates the improvement on the safety that a pattern causes. The safety of a pattern is assigned to one of the safety quantifier S1, S2, and S3.

Cost This aspect gives the implications on costs of a pattern, which includes the recurring cost per unit and development cost of the pattern. The cost of a pattern is assigned one of the cost quantifiers C1, C2, and C3.

Modifiability This aspect indicates the degree to which a system developed according to a pattern can be modified and changed. The modifiability of a pattern is assigned to one of the modifiability Quantifiers M1, M2, M3, and M4.

Table 5.2: Description of the quantifiers of the quality attributes

Quality Attribute	Quantifiers	Description
Reliability	R1	Lower than a basic system
	R2	The same as a basic system
	R3	Higher than a basic system
Safety	S1	Lower impact
	S2	Small improvements
	S3	Incremental improvements
Cost	C1	High cost
	C2	Reasonable cost
	C3	Low cost
Modifiability	M1	Can be modified with extra effort
	M2	Can be modified with easy steps
	M3	Very simple to be modified
	M4	The same as a basic system
Execution time	T1	Increase in the execution time
	T2	Little influence
	T3	No effect

Impact on Execution Time This aspect shows the effect of a pattern on the total time of execution at runtime. The impact on execution time of a pattern is assigned to one of the impact Quantifiers T1, T2, and T3.

5.4.2 Comparison Results and Discussion

Table 5.3 shows the comparison results of all the selected patterns and our pattern. This table summarizes the knowledge we gathered on architecture patterns and quality attributes. We assigned quantifiers to each quality attribute of the patterns based on the analysis in our preliminary literature study.

From the reliability perspective, only Triple Modular Redundancy pattern can improve the reliability of a basic system. The reason is that this pattern can continue to work correctly as long as two or more channels have no fault. The Safety Channel pattern has no significant effect on reliability.

From the safety perspective, the Triple Modular Redundancy pattern leads to the highest number of safety improvements, while the Safety Executive pattern has the lowest number of safety improvements. The reason is, the safety improvement of Triple Modular Redundancy pattern is equal to the relative reliability improvement due to the redundancy in the pattern. However, the safety improvement of Safety Executive pattern relies on the reliability and coverage factor of the Safety Executive component as well as the reliability of the fail-safe Processing Channel. In the Safety Channel pattern, the safety improvement depends on the reliability and the coverage of the safety channel as well as the reliability of the Limp Home channel. The coverage of the Safety Channel is normally higher than the coverage of Safety Executive component in the Safety Executive pattern. Thus, we conclude that Safety Channel pattern brings higher safety improvements than the Safety Executive pattern.

From the cost perspective, we can see that among these patterns, Monitor-Actuator,

Table 5.3: Results of comparison with other patterns

Patten Name	**Reliability**	**Safety**	**Cost**	**Modifiability**	**Impact on Execution Time**
Triple Modular Redundancy	R3	S3	C1	M4	T2
Monitor-Actuator	R2	S2	C3	M2	T3
Sanity Check	R2	S1	C3	M3	T3
Safety Executive	R2	S1	C2	M2	T3
Protected Single Channel	R1	S2	C3	M1	T2
3-Level Monitoring	R2	S2	C3	M2	T1
Safety Channel	R2	S2	C2	M2	T3

Sanity Check, Protected Single Channel, and 3-Level Safety Monitoring pattern are low-cost patterns. The other three are more costly to realize. Triple Modular Redundancy pattern is costly due to a high recurring cost of using three parallel models, while Safety Executive and Safety Channel pattern are costly due to the development cost of three different channels.

From the modifiability perspective, the Triple Modular Redundancy and Sanity Check patterns are easier to modify than the other four patterns. The Protected single-channel pattern does not change the modifiability level of the basic system. Sanity Check pattern only requires little extra work comparing to the basic system. As for Safety Channel, it is relatively easy to add new functionality to the system. Simple steps are needed: adding new functionality as a new channel, updating the failure model of Health monitor, and ensuring the sufficiency of the Limp Home channel.

Finally, from the impact on execution time perspective, only 3-Level Safety Monitoring pattern causes big influence. This influence is because the total execution time of this pattern is affected by the time to execute some components or modules in its three levels. However, for Safety Channel, there is no impact on execution time due to the usage of different CPUs for each channel.

The comparison results of these patterns can be different if they are compared from different perspectives or in different contexts. Therefore, when engineers choose which pattern to be used or applied, they could make the comparison more specific in the context of their use scenarios, which can be facilitated by using the Pugh method.

5.5 Validation Based on Case Study

The EcoTwin project is an effort towards higher levels of automation for trucks. The application in this project is truck platooning. In this project, a few donor trucks are equipped with additional systems to realize the target application. Realization of this application requires access to various sensors and actuators in donor trucks as well as additional sensors/actuators such as a vehicle to vehicle communication. The three main actuators are the engine (power train), the brake, and the steering. Therefore, the full functional architecture of the system has two actuator channels: a longitudinal, and a lateral channel.

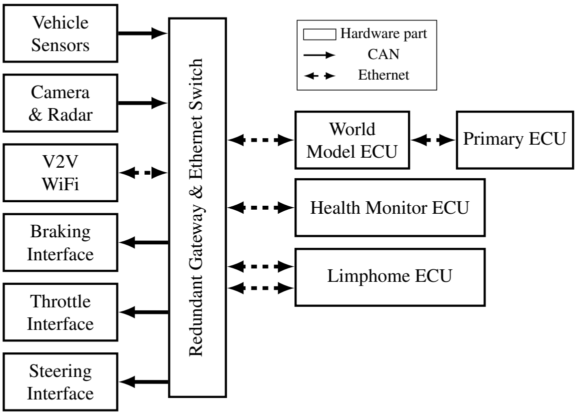


Figure 5.5: The (simplified) hardware view of EcoTwin system

We have been busy implementing the proposed pattern on the functional view for its level 3 truck platooning application [9] in the EcoTwin project. We faced two major challenges concerning the design: the mapping of functionality to hardware platforms, and the realization of fault containment for each channel. Fault containment means that in case of a failure, the fault does not propagate. Thus no incorrect information is sent to other components in the system. Fault containment prevents faulty information to be transmitted and thus preventing actuation based on faulty signals. Modifiability of the proposed pattern brings benefit by allowing the Health monitor and Limp home channels to be shared between the two actuator channels for longitudinal and lateral control.

The hardware architecture view of the EcoTwin system is shown in Figure 5.5. The choice for network topology and technology is based on the latency and throughput requirements of the application. Therefore, an Ethernet network is used for internal communication for the added systems. Practical considerations resulted from the legacy of the donor truck created some challenges for integration; particularly, creating the required redundancy considering (CAN based) bus topology of the donor truck, was challenging. The gateway technology used for converting CAN to Ethernet is duplicated for the safety-related signals. Note that in this chapter, we did not show the functional architecture of the implementation as it is similar to the patter, but for more details you can refer to our other publication [8].

Moreover, there is redundant Ethernet communication from the Limp home ECU to the gateway to provide robustness against hardware failures of Ethernet switch. In order to provide sufficient fault containment, each channel is mapped to separate hardware. An exception is the actuator channel; mapping of the two actuator channels to a single ECU was not possible due to required computational power. Therefore, to increase efficiency, the input processing, and data integrity check are mapped to one, and d ata transformation and output processing are mapped to another ECU.

5.6 Conclusions

In the context of automated driving, safety becomes even more crucial. When developing safety-critical functions for automated driving, architecture patterns are highly recommended as a solid basis for the system architecture. In this chapter, we presented a novel architecture pattern (Safety Channel) suitable for the architecture of automated driving applications. This pattern is based on the Safety executive pattern and inspired by the E-Gas architecture. A generic approach to compare the proposed pattern with some selected patterns, according to several quality attributes, has been demonstrated. The comparison result can be used as a basis for architecture decision making process in a specific project.

Chapter 6

On the Impact of Early Design Decisions on Quality Attributes of Automated Driving Systems

Initiatives such as smart mobility and automated driving bring new concerns such as safety and security for the automotive industry. New architectures and designs are required for the in-vehicle systems to address these emerging concerns. Early design decisions have a large impact on the required functionalities as well as the quality attributes of these systems. Understanding the impact of design decisions on the quality of the system is crucial for successful system development. It is difficult to predict the requirements for safety (at an early development stage) considering the innovations of automated driving. These safety requirements have a considerable influence on project planning and development cost. Therefore, it is important for the industry to understand the decision points and their impact on system design. In this chapter, we share our experience with applying architectural patterns to automated driving systems. We particularly discuss the impact of design decisions regarding the operational design domain on (functional) safety. We provide two automated driving systems as discussion cases and investigate the impact of the operational situation on the safety requirements such as safe state and degraded operating mode. We show how posting small constraints on the operational situation can result in the simplification of the sensor and actuator requirements of these systems.

This chapter is based on:

[68] A. Khabbaz Saberi, J. Vissers, F. P. A. Benders, "On the Impact of Early Design Decisions on Quality Attributes of Automated Driving Systems," *13th Annual IEEE International Systems Conference (SysCon 2019)*, April 2019, Orlando, Florida, USA.

6.1 Introduction

The automotive industry is seeing massive innovations in recent years in the context of smart mobility and automated driving [88], [89], [130]. These innovations result in changes in the automotive systems in the form of new architectures and designs as well as new technologies. As an example, the platooning application for trucks [93], [102] requires new communication technologies. New architectures and design paradigms have been subject of research in this domain [8], [64], [79]. The success and public acceptance of these new systems depend on their quality aspects such as safety and security.

Early design decisions have a great impact on the required functionalities as well as quality attributes of systems. These topics are captured in the ISO/IEC/IEEE 42010:2011 [57]. This standard is the basis of many research articles that shed light on the topic of design decisions [97], [100]. However, from a practical perspective, we need quite some information regarding the requirements for the quality of systems in the definition phase of development projects.

In this chapter, we investigate early design decisions for two cases. We particularly discuss the impact of decisions regarding the operational situation on functional safety (as a quality attribute). Our contribution is to provide two industrial case studies for tracing the impact of early decisions on the functional safety requirements.

The remainder of this chapter is organized as follows: In Section 6.2, we provide some background information regarding automated driving, operational design domain, and safety of automated driving, and share our viewpoint on these topics. In Section 6.3, we share two industrial cases on automated driving applications. We use these cases for discussion on the impact of early design decisions on the architecture from the safety viewpoint in Section 6.4. Finally, Section 6.5 concludes this chapter.

6.2 Background Information

In this section, we provide background information as well as our view on three topics: Automated Driving (AD), Operational Design Domain, and safety of AD.

6.2.1 Automated Driving

Automated Driving (AD) is one of the most disruptive technologies of recent years. The expectation is that it will have a great social and economic impact [88]. From a development perspective, AD systems can be seen as an advancement of Advanced Driver Assistance System (ADAS). The SAE J3016 [111] defines the six levels of automation where the Dynamic Driving Task (DDT) is gradually shifted from the driver to the systems in the vehicle. Systems that provide automation level above SAE Level 3 are considered AD systems. In this chapter, we use these levels as the reference for the automation levels.

In all automation levels, the system is responsible for the functional safety of the internal systems within the vehicle. Starting from the SAE Level 3, the AD systems are expected to monitor the driving environment and the response to objects and events; whereas, the driver is still in charge of this for vehicles equipped with systems below SAE Level 3. This demand on monitoring creates the main challenge for AD systems regarding the environmental situation awareness. The difference between SAE Level 4 and Level 5 is that the automation for Level 4 is limited to specific Operational Design Domain(s) (ODD),

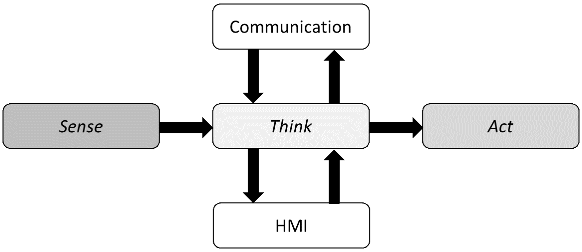


Figure 6.1: High-level functional architecture

whereas Level 5 is not conditional anymore. Therefore, a Level 5 vehicle is expected to always perform the DDT instead of the driver.

We can identify the main functionalities of AD systems within the "*sense*, *think* and *act*" paradigm [118]. For an AD system, *sense* refers to all the functionalities required for gathering data from the environment and vehicle, and processing this data into useful information for control of the vehicle. This data includes all the sensory inputs as well as processing and sensor fusion technologies. *Think* can be seen as the functionalities that perform planning, decision making and control of the vehicle using the information from sense; we can find examples in all the various levels of control from the highest tactical decision functionalities such as path planning, to the lower level controls which deal with vehicle dynamics such as longitudinal/lateral control. *Act* refers to all the functionalities in the vehicle that interact with the environment to control the vehicle; these activities are performed by electronics and machinery instead of a human driver.

In our view, two additional functionalities are required for an effective AD system design: *Communication*, and *Human Machine Interface (HMI)*. These two functions may be considered as part of the sensing or actuation; however, due to special characteristics of these two functions, it is worthwhile to consider them separately. The communication component is about information exchange between the vehicle and infrastructure or other vehicles (V2X/V2V). For this function, quality aspects such as safety and security need special attention. Performance indicators such as latency are crucial for functional safety and capabilities of an AD system. The HMI provides the functionality for interacting with the driver or passengers. This interaction includes presenting information to them and receiving input/commands (including inputs for DDT). This component plays a major role for quality as well. From the functional safety point of view, the information presented to the driver during degraded operating mode (and eventually transfer of control) is important. Moreover, reasonably foreseeable misuse of the system should be considered for HMI related functionalities. Considerations derived from analysis of reasonably foreseeable misuse have major impact on the ability of the designed functionality for a safe transfer of control (relevant for up to SAE Level 4) and achieving the assumptions about the roles of involved people in an emergency operation (relevant for up to SAE Level 5). Figure 6.1 shows the interaction among these high-level functionalities.

6.2.2 Operational Design Domain

Operational Design Domain (ODD) is defined by [111] as:

Definition 7. *"Operating conditions under which a given driving automation system, or*

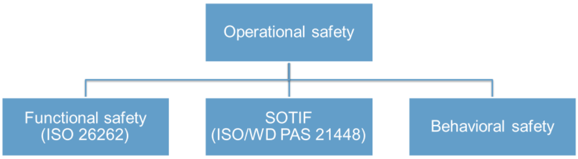


Figure 6.2: Operational Safety has three parts: Functional Safety, SOTIF, and Behaviour Safety.

feature thereof, is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics."

In short, ODD is the a model of the environment under which the system must function correctly. As such, the definition of the ODD is crucial for understanding the automation level up to SAE Level 4. Per definition, Level 5 is not limited to any ODD, meaning that the system must operate in any condition. We expect that ODD helps with the specification, assessment, testing, and validation requirements of an AD system, as the ODD specifies the conditions, limitations, and the intended use of the system. Some examples of ODD are Closed environment, Highway, speed $> 60km/h$, and City driving speed $< 70km/h$.

6.2.3 On the Safety of Automated Driving

Better safety is among the most important motivations for the developments towards higher automation levels. The AD systems make the traffic safer by removing the driver, who is responsible for more than 90% of traffic accidents [92]. The overall safety of an automotive system has multiple aspects. The ISO 26262 standard covers functional safety, which addresses safety in case of a failure. ISO PAS 21448 [55] captures the safety concerns regarding the correct intended functionality. These two standards capture most of the safety concerns when it comes to vehicle development. However, on the traffic level, we need to also think about traffic behavior and what we call behavioral safety. This aspect captures concerns such as mixing automated vehicles with manually driven vehicles. An overview of these aspects is shown in Figure 6.2.

6.3 Automated Driving Application Examples

In this section, we provide two examples of automated driving applications. We provide these examples from TNO Automotive project portfolio[1]. We use these examples to discuss the architectural decision-making process and impact in Section 6.4. We describe the platooning application and the Driverless Off-road application. We describe three factors for each application: First, in the system context, we identify the external systems or entities that interact with the application. Second, we describe the high-level functional architecture of the system. Finally, we describe the safe states of the system. A safe state is an operating mode or a maneuver in case the application is not able to perform the nominal

[1] The descriptions provided here are limited to non-confidential project information.

functionality according to the specification. Safe states are part of Safety Goals resulting from Hazard Analysis and Risk Assessment (HARA) according to ISO 26262 [54].

In Section 6.4, we discuss the resulting functional safety architecture (part of the Functional Safety Concept) and the requirements regarding Automotive Safety Integrity Level (ASIL) of the sensor elements.

6.3.1 Platooning Application

The Platooning application provides a conditional automated driving (SAE level 3) with the speed range of up to 90 km/h on highways/motorways or similar roads. This application operates from the entrance to the exit of the highways on all lanes. It provides a range of functionalities in both longitudinal and lateral control, including overtaking. It provides a system in which multiple trucks communicate acceleration/braking actions as well as steering intentions via Vehicle to Vehicle (V2V) wireless communication.

6.3.1.1 System Context

The platooning application provides value in a few ways: reduced fuel consumption through a reduction of aerodynamic drag and increased comfort by smoothening of speed variations induced by traffic, as well as increased road capacity. The Platooning application interacts with the driver, other traffic participants, the environment (e.g., visible lane markers and weather conditions), and the infrastructure (e.g., speed limits and traffic light status).

6.3.1.2 Functional Architecture

The application description (including ODD) and the system context are used to derive the functions as needed to perform the Platooning application. A schematic diagram of the functional architecture and interactions is shown in Figure 6.3. As can be seen in this architecture, all sensor information is collected and processed first by the World-modeling function such that a consistent picture of the vehicle and the outside world is formed for all functions in the architecture. This World-modeling function distributes the fused and processed data to the Planning, Decision, and Vehicle-control functions. The World-modeling function also shares information via V2X communication. This information enables synchronized control across the platoon members. The Control functions communicate to the HMI Interaction function. The HMI Interaction function manages the data of the HMI Display function. The cellular communication function provides information to the Planning function. All the V2X communication functions interact with the World-modeling, Decision, and Vehicle-control functions. The only function that communicates to the Actuator functions is the Vehicle-control function to ensure a singular input for actuator control. The Vehicle-control function controls the output to the actuation functions. The Planning function communicates with the external systems via the Cellular function.

6.3.1.3 Safe State

The application can use degraded functionality to make the transition to the safe state. The safe states can be derived for the Platooning L3 applications.

1. Hand over control to the driver in case appropriate time is available for this transition of control.

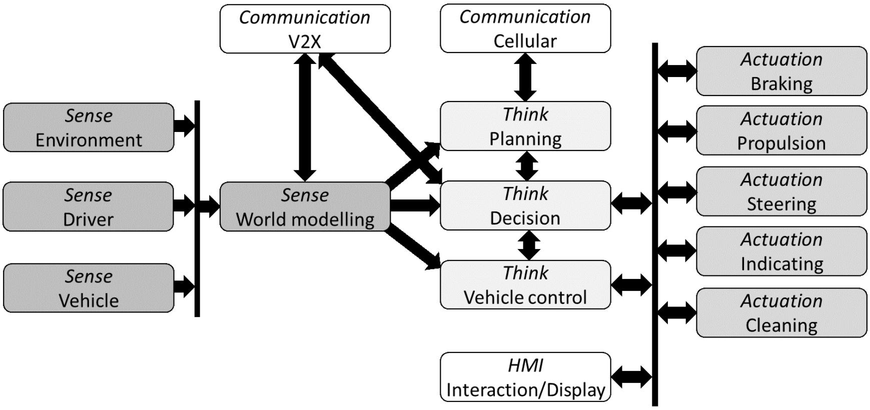


Figure 6.3: The functional architecture of the Platooning application

2. Decelerate in lane, dependent on pre-warning of the system, to avoid hazardous situations.

3. Maneuver to the emergency lane (or rightmost lane if no emergency lane is available) and decelerate to a standstill.

6.3.2 Driverless Off-Road Application

The Driverless Off-Road Level 4 application supports full automation in a confined area with off-road, paved and gravel surfaces driving up to a speed of 50 km/h. This environment differs from the other applications in the sense that there are no fixed infrastructural elements like asphalt roads with lane markers. A possible application scope would be an automated tipper truck in an opencast mining environment.

6.3.2.1 System Context

Multiple automated vehicles in this application are monitored and controlled remotely by a central control system, which is controlled by a fleet manager. The fleet manager communicates task (directions, routes, and speed restrictions) to the vehicles. The vehicles have to achieve these tasks including moving payload from one location to another location using several functions. A central control system monitors the state of the fleet. This system has the option to overrule the automated vehicle at any time (e.g., stop complete operation) for safety reasons and risk mitigation.

The responsibilities of automated vehicles are to perform the directed task by navigating through the environment and communicating their status (position, velocity, vehicle performance, sensor performance) to the central control system. Besides, automated vehicles have to avoid collisions with other vehicles and obstacles in the environment. Moreover, these vehicles have to park accurately in a predefined orientation, at a specified location.

There is no driver in the cabin of the vehicles. In case of maintenance or external transport, the vehicle moves to a safe location and shuts down the system. Then, a driver can step into the vehicle.

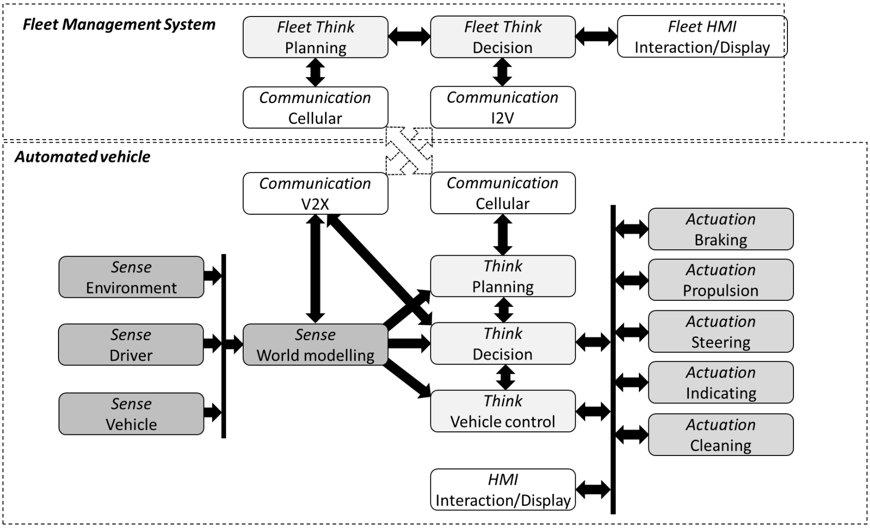


Figure 6.4: The functional architecture of the off-road driverless application

The external systems that interact with this application are the central computer, the driver (only possible and allowed in specific conditions), the environment (e.g., road blockages, guard rails), and the infrastructure or other vehicles (e.g., systems using I2V communication to support the vehicle to localize).

6.3.2.2 Functional Architecture

The high-level functions in the vehicle are similar to the functional architecture of the Platooning application. The only difference is that the application in the vehicle has to communicate to the central control system (fleet manager) to receive tasks and goals to execute. This information is exchanged via I2V or Cellular communication.

The V2V communication is used to exchange information between the vehicles to avoid collisions. The HMI and Sensing Driver functions are only used in case the vehicle is manually driving. The functional architecture of this application is shown in Figure 6.4.

6.3.2.3 Safe State

The safe state for this application is relatively simple. The vehicle has to decelerate and stop driving (park the vehicle) and notify the status to the central control computer (fleet manager).

In the case of a severe (system) failures, the application should park the vehicle and turn off the engine. Furthermore, the central control system should ensure the safety of the environment to the failed vehicle by redirecting other automated vehicles. Another option is that the central computer remotely controls the vehicle when possible. Since failing vehicles still need to be removed from the environment, maintenance operators should be able to get to the failed vehicle and move it; this may require communication with other operational automated vehicles. In case of less severe failures, the application may use degraded functionalities (like reduced speed) to continue their operation or move to maintenance zones.

In case the communication to the central computer is failing, the vehicle also has to notify this failure and park the vehicle. In this case, the central computer also notifies this failing communication. In case the vehicle is transitioning to the safe state, the central computer can redirect the other vehicles in the environment to avoid collisions. The central computer can also avoid potential collisions by ensuring that the vehicles are never too close to each other by allocating the vehicles in safety zones or reducing the vehicles velocities when they have to operate close to the other zones.

We summarize these safe states as follows:

1. Hand over control to the central control system in case appropriate time is available for this transition of control.

2. Decelerate in (virtual) lane, dependent on pre-warning of the system, to avoid hazardous situations.

Note that the compared to the other application (Platooning Level 3 application) there is one fewer safe state. Maneuvering to the emergency lane is not required as infrastructural does not include lane markings. Potentially, this safe state could be adapted for off-road application to "maneuvering ot a designated safe area."

6.4 Discussion

In the previous section, we presented the functional architecture excluding the safety measures. The safety concept (ISO26262 & SOTIF) leads to new functional (safety) requirements for the systems. These safety measures have an impact on the functional architecture of these applications. The impact on the control and actuating functions is described in Chapter 5. Here, we focus on the environmental sensing function. We assume that the lane information is available and the AD applications can distinguish the Left, Ego, and Right Lanes. For the case of the Off-road application, we assume to have access to similar information indicating "*virtual*" lane information. Ego Lane is the lane in which the vehicle is located at a given time; the Left and Right Lanes are, respectively, the lanes to the right and the left of the vehicle. We analyze the ASIL ranking of these sensors by range and area of detection; and demonstrate that our two example applications have different safety requirements regarding their sensors, even though the nominal functional architecture of the two is fairly similar on the vehicle side.

We show an overview of the required environmental sensing functionality for Platooning Level 3 in Figure 6.5. The short, medium and large areas for the sensors are depicted together with the ASIL classification.

The sensors dedicated to the forward detection range in the EGO lane (the vehicle driving lane) have a higher ASIL for the short and medium range to achieve the collision avoidance requirements. Moreover, the sensors for detection on the right lane (for all ranges in the forward and side direction) are essential for transitioning to the third safe state of the Platooning Level 3 (maneuver to the emergency lane). Therefore, these sensors are ranked at ASIL D.

The required environmental sensing functionality for Driverless Off-Road Level 4 is shown in Figure 6.6. As can be seen, the sensors for the forward detection range on the (virtual) EGO lane have an ASIL D only for the short range. This difference (with respect to Platooning) is resulting from the difference in the speed range requirements of the ODD.

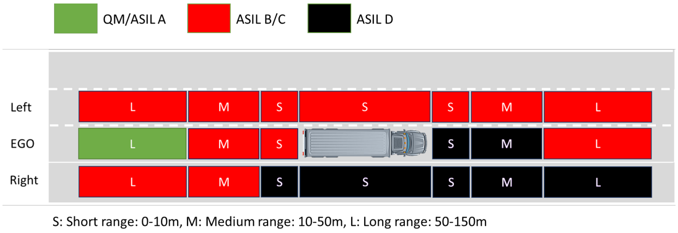

QM/ASIL A
ASIL B/C
ASIL D
Left
EGO
Right
L
M
S
S
S
M
L
S: Short range: 0-10m, M: Medium range: 10-50m, L: Long range: 50-150m

Figure 6.5: Platooning Level 3 sensing requirement and ASIL levels

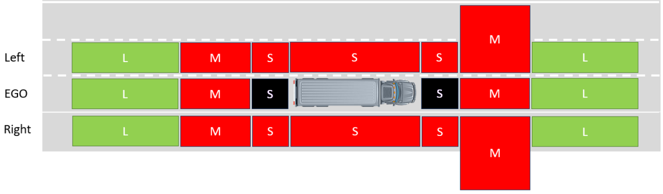


Figure 6.6: Driverless Off Road Level 4 environmental sensing requirement and ASIL levels

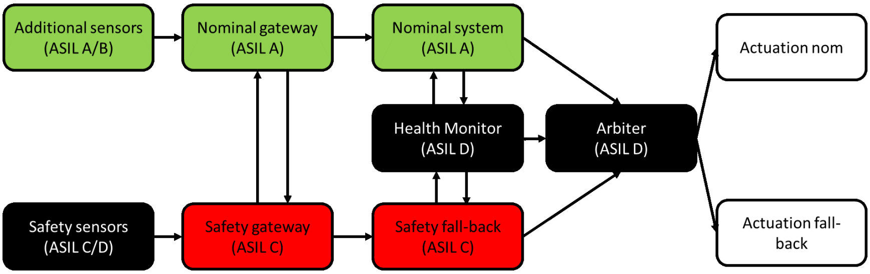


Figure 6.7: Extended Safety Channel pattern for Platooning Level 3 and Driverless Off-Road

As the required range, the detection areas depend on the maximum operating speed of the vehicles. The (virtual) Ego Lane rearward detection also has an ASIL D ranking to account for the more frequent use of rear gear on the Off-Road ODD. Furthermore, since safe states of this application do not require vehicle maneuvers to other (virtual) lanes, there are no other ASIL D requirements for the (virtual) Right Lane sensors. Because of the same reason, the ASIL ranking of the steering actuators is also lower for the Driverless Off-Road (compared to Platooning).

We use an extended version of the Safety Channel pattern as introduced in Chapter 5 for both applications to extend the functional architecture and arrive at the functional safety architecture. Figure 6.7 shows a schematic of this extended pattern.

The safety sensors and safety gateway are required for the monitoring of the environment and triggering the transition to a safe state. The safety fall-back functionalities are backup for transition to a safe state in case of a system failure. Note that in this pattern all sensors are used for the nominal functionality. If we would make a complete decoupling

Table 6.1: Mapping of the elements of the pattern to functions of application examples

Element of Pattern	Mapping of Platooning	Mapping of Off-Road
Additional Sensors (ASIL A/B)	*Sense* Environment, *Sense* Driver, *Sense* Vehicle	*Sense* Environment, *Sense* Driver, *Sense* Vehicle
Safety sensors (ASIL C/D)	*Sense* Environment (covering the area indicated in Fig 6.5)	*Sense* Environment (covering the area indicated in Fig 6.6)
Nominal gateway (ASIL A)	(N/A) Note that interactions are abstracted in the functional architecture presented here.	(N/A)
Safety gateway (ASIL C)	(N/A)	(N/A)
Nominal system (ASIL A)	*Sense* World modeling, *Communication* V2X, *Communication* Cellular, *Think* Planning, *Think* Decision, *Think* Vehicle control, *HMI* Interaction/Display	*Sense* World modeling, *Communication* V2X, *Communication* Cellular, *Think* Planning, *Think* Decision, *Think* Vehicle control, *HMI* Interaction/Display
Safety fall-back (ASIL C)	*Think* Decision and *Think* Vehicle control	*Think* Decision and *Think* Vehicle control
Health Monitor (ASIL D)	(N/A) Note that the functional architecture does not include safety measures	(N/A)
Arbiter (ASIL D)	(N/A)	(N/A)
Actuation nom	*Actuation* Braking, *Actuation* Propulsion, *Actuation* Steering, *Actuation* Indicating, and *Actuation* Cleaning	*Actuation* Braking, *Actuation* Propulsion, *Actuation* Steering, *Actuation* Indicating, and *Actuation* Cleaning
Actuation fall-back	(N/A)	(N/A)

and not use these safety sensors for the nominal system, then we would need the additional sensors to cover the full range around the vehicle, including the range as already covered by the safety sensors. The aforementioned extension to the Safety Channel pattern is to use two sets of sensors: a nominal set, and a safety set of sensors. The additional safety sensors (which only cover the area that is not yet covered by the safety sensors) are routed via a nominal and an additional safety gateway. The consequence of this decoupling is that only the safety sensors need to comply with the higher ASIL requirements. We provide the overview of the mapping of the functions as described in the functional architecture in the examples to the elements of the proposed pattern in Table 6.1.

The safe states (and the transition to safe state) for Driverless Off-Road is simpler than the Platooning Level 3. Even though the former is a Level 4 application. The simplicity entails that the safety fall-back function for Driverless Off-Road can be achieved by "braking" only and the safety sensors require a lower coverage area. Therefore, the Actuation Fall-back function suffices for the transition to a safe state. The communication to the central system is implemented in the Actuation Fall-back function. Therefore the ASIL levels can be decomposed into the communication system and central control system that become more responsible for the safe operation on the Driverless Off-Road.

6.5 Conclusions

In the previous chapter, we discuss how two early design decisions, namely: the choice of the ODD and the safe state impact the complexity of the safety fall-back functionality and the requirements on the detection range of the safety sensors. Even though for both of examples we use the Extended Safety Channel pattern to design the functional safety architecture, the ASIL ranking and reliability requirements of the safety fall-back functionality remains dependent on the decisions regarding safe states. We can reduce the safety requirements on the sensors by choosing a simpler safe state (e.g., parking in the EGO lane).

We suggest using the extended Safety Channel pattern for designing the functional safety architecture. This pattern requires the addition of a health monitor and arbiter to the design. These two functions have a higher ASIL and reliability requirements. The development challenge for both cases is the Health Monitor due to the dependencies of this function to the implementation. Since Health Monitor has to check the failures on both functional and implementation (Hardware and Software) levels; therefore, as future work, we would like to research on methodologies for automatic health monitor generation.

Chapter 7

On Functional Safety Methods: A System of Systems Approach

Connectivity plays a crucial role in enabling automated vehicles to navigate, as well as in regulating this newly established network of connected vehicles as efficiently and safely as possible. As a result, modern vehicles are equipped with Vehicle to Vehicle (V2V) and Vehicle to other systems (V2X) communication capabilities. Vehicles, traditionally considered as a *monolithic* system, now become part of an ecosystem of vehicles, infrastructure and mobility services that can be characterized as a System of Systems (SoS). The SoS aspect requires novel safety methods that are applicable. In this chapter, we investigate the impact of applying safety analysis to a SoS with a conventional, "vehicle-centric" development process. We propose a tailored safety lifecycle based on guidelines of ISO 26262 that is augmented to encompass additional considerations pertinent to a SoS. We performed a comparative study by applying our proposed method as well as the traditional (vehicle-centric) approach as per ISO 26262 for safety engineering of a truck platooning application. The comparison results show the overall effectiveness of the proposed method. The "connected vehicles" development process resulted in more safety goals compared with the vehicle-centric approach. This increase may, at first thought, suggest that this approach requires a significant effort increase as the number of safety goals is an indicator of the amount of needed effort for the safety engineering process. However, the safety analysis (e.g., fault tree analysis) of the platoon system from a vehicle-centric approach exponentially grows in size. This increase in complexity of analyses means that the actual effort required of the proposed method for the SoS is comparatively more efficient. Besides, the proposed comparison showed us that the resulting safety analyses from our suggested method, in particular, the Fault Tree Analyses (FTA) are less prone to error thanks to less complexity in the FTA graphs. Creating an appropriate level of abstraction for the vehicle and the platoon makes the analysis more effective.

This chapter is based on:

[64] A. Khabbaz Saberi, E. Barbier, F. Benders and M. G. J. van den Brand, "On functional safety methods: A system of systems approach," *12th Annual IEEE International Systems Conference (SysCon 2018)* April 2018, Montreal, Quebec, Canada, p. 261-267

7.1 Introduction

The automotive industry is moving towards highly automated and connected vehicles. Achieving Level 4 and Level 5 of automation [113] entails development of connected vehicles and a significant improvement in functional safety. Recent research on this topic is performed on the integration of functional safety in the development process for automated driving [66], [79]. Connected vehicles depend on the interaction and reaction of the involved vehicles and rely on the exchanged information between the vehicles through Vehicle to Vehicle (V2V) or Vehicle to other systems (V2X) communication. Applications based on connected vehicles provide an overall traffic behavior that can be seen as a System of Systems (SoS). An example of a connected vehicles application is truck platooning, which is also a good example of safety impact as the collective behavior entails new considerations on safety [93].

Systems of systems have a few major characteristics that differentiate them from Monolithic Systems (ML) [75]. A SoS consist of individual elements (e.g., vehicles, fleet management, traffic infrastructure) that have an independent life cycle and evolve (and are managed) independently. Furthermore, most of the elements of a SoS operate independently, i.e., they have meaningful functionality that provides value to their stakeholders; the stakeholders may or may not be the same for the SoS (e.g., city authority, government) and its elements (e.g., road users, taxi company). In addition to the individual functionalities, SoS often exhibits emergent behavior that benefits both the SoS stakeholders (e.g., increased traffic safety) as well as the SoS elements stakeholders (e.g., faster journey, lower levels of pollution). Other characteristics usually include geographical distribution and heavy dependency on network technologies.

Concerning safety ISO/DIS 26262:2018 [54], the current industry state of the art, defines an Item as:

Definition 8. *"system or **array of systems** to implement a function or part of the function at the vehicle level, to which ISO 26262 is applied."*

This definition allows development of a System of Systems per this standard. However, if the boundary of SoS extends over several vehicles, the safety analysis methods as proposed by this standard are no longer capable of capturing all possible safety challenges. Ensuring the safety of a SoS defined across several connected vehicles or including other types traffic systems, requires analysis at traffic level as well as vehicle level.

Similar studies have addressed this challenge. A model-driven approach for capturing the dependent failures of elements of a SoS is given by [20]. A new method for SoS hazard analysis is proposed in [5],where a systematic approach to hazard identification is introduced. In this chapter, we address the topic of safety for SoS in the context of automotive applications. We tailored and extended the ISO 26262 safety lifecycle by adding required activities for SoS safety analysis. Our approach is relatively similar to [5], but we further study the impact of the proposed method on subsequent safety activities by a comparative study. Besides, we examine the impact of our proposed "connected vehicles" approach on the effort required for functional Fault Tree Analysis (FTA) and system validation. In this research, we used both the traditional vehicle-centric approach as well as the proposed "connected vehicles" approach for the development of the same platooning system. The scope of our research is limited to what ISO 26262 refers to as "the concept development" phase. This phase includes the Item definition, the Hazards Analysis and Risk Assessment (HARA), Functional Safety Concept (FSC) and System Verification and Validation

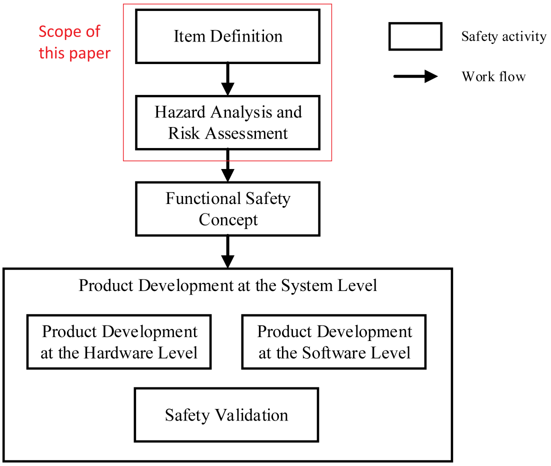

Scope of this paper
Item Definition
Hazard Analysis and Risk Assessment
Functional Safety Concept
Product Development at the System Level
Product Development at the Hardware Level
Product Development at the Software Level
Safety Validation
Safety activity
Work flow

Figure 7.1: Simplified safety lifecycle during the concept and system development phases recommended by ISO/DIS 26262

at the vehicle level.

The rest of this chapter is organized as follows: In Section 7.2, we describe the proposed method and required activities. The comparative study is explained in Section 7.3, and the discussion on the results is given in Section 7.4. Finally, Section 7.5 concludes this chapter.

7.2 The Proposed Tailored Safety Lifecycle

The safety lifecycle as described by ISO/DIS 26262 starts with the Item Definition. This step provides the required information about the item to perform the rest of activities in the safety lifecycle; this includes information such as the item intended functionality, intended use, operational situations, required operating modes, and the preliminary functional architecture. The Hazard Analysis and Risk Assessment (HARA) is the immediate step after the Item Definition in the concept phase. By performing a HARA, the hazards related to the item are systematically identified and categorized. Furthermore, safety goals are defined to avoid those hazards. The next activity is the Functional Safety Concept (FSC). The FSC results in a system level (functional) architecture that is designed to satisfy safety requirements. The starting point of FSC is the analysis of the potential item's failure modes that (may) violate the safety goals. Once the pertinent failure modes of the various system elements are identified, safety measures (formulated as Functional Safety Requirements (FSRs)) are defined. The FSRs are subsequently allocated to the relevant architectural elements. This phase is followed by product development at the system level, which includes: Software and Hardware Level Development as well as Safety Validation. A model of the concept and development phase of the safety lifecycle as described by ISO 26262 is depicted in Figure 7.1.

Our proposed tailored safety lifecycle (per ISO 26262-2 5.4.6: project independent tailoring of the safety lifecycle) is shown in Figure 7.2. In this method, the decomposition into various abstraction levels is initiated at the very first step of the functional safety development process: the item definition. The functions, system definition granularity, and

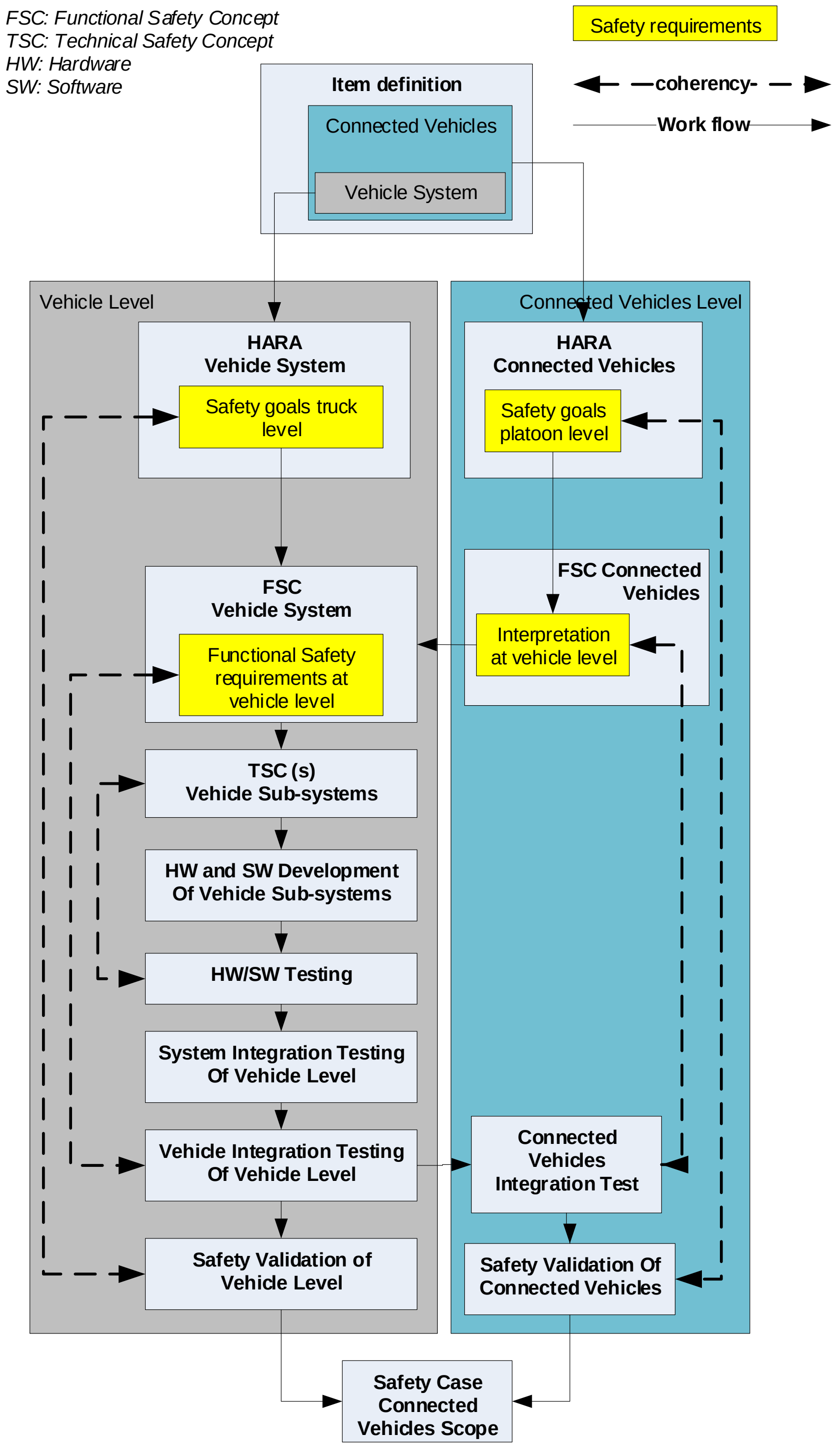

FSC: Functional Safety Concept
TSC: Technical Safety Concept
HW: Hardware
SW: Software
Safety requirements
coherency
Work flow
Item definition
Connected Vehicles
Vehicle System
Vehicle Level
Connected Vehicles Level
HARA
Vehicle System
Safety goals truck level
HARA
Connected Vehicles
Safety goals platoon level
FSC
Vehicle System
Functional Safety requirements at vehicle level
FSC Connected Vehicles
Interpretation at vehicle level
TSC (s)
Vehicle Sub-systems
HW and SW Development Of Vehicle Sub-systems
HW/SW Testing
System Integration Testing Of Vehicle Level
Vehicle Integration Testing Of Vehicle Level
Connected Vehicles Integration Test
Safety Validation of Vehicle Level
Safety Validation Of Connected Vehicles
Safety Case Connected Vehicles Scope

Figure 7.2: The proposed tailored safety lifecycle

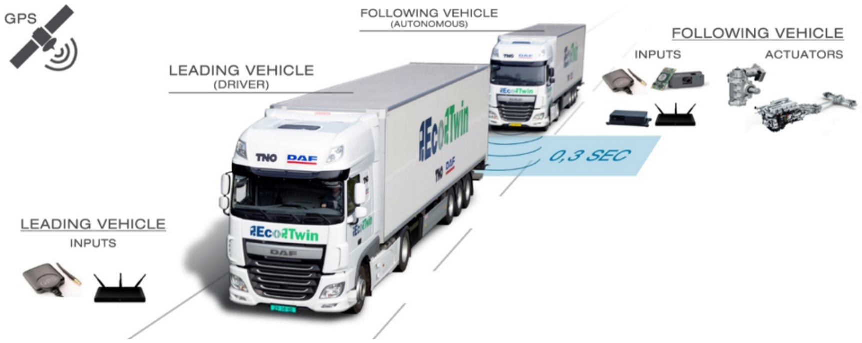


Figure 7.3: EcoTwin truck platooning concept

use cases are defined at the vehicle and “connected vehicles” levels.

Similarly, in parallel to the vehicle level hazard analysis, a SoS hazard analysis is performed aiming at identifying the hazards that may result from the overall emergent behavior. Then, again in parallel to the vehicle level FSC, a safety concept for “connected vehicles” level is developed to discover any required safety measure that needs to mitigate hazards at the “connected vehicles” level but effectively implemented at the vehicle level. This activity is supported by safety analyses such as Fault Tree Analysis (FTA). The resulting safety measures are then consolidated with the vehicle level FSC to ensure the allocation of these requirements to system level technical safety requirements. From this point, hardware and software development can be conducted conventionally per parts 5 and 6 of ISO 262626. Once the integration and safety verification at the truck level is conducted, an additional level of safety verification level is required to test the correct execution of the safety mechanisms at the “connected vehicles” level. These two levels of verification imply two levels of safety validation (ensuring respectively the ability of the implemented safety mechanisms to mitigate the hazards at the vehicle and “connected vehicles” levels).

7.3 Comparative Study

In this section, we describe the performed comparative study for evaluating the proposed method. First, we explain the project context as it is the same for both analysis approaches. Then, we describe the safety analysis by applying our proposed SoS approach. We follow that with the Vehicle-Centric approach safety analysis. Finally, we provide some remarks on the System Level Development and Safety Validation of the platooning system.

7.3.1 EcoTwin Context

The EcoTwin project is an effort towards higher levels of automation (Levels 3 and 4) for truck platooning with the goal to create a platooning concept, which can reduce fuel consumption, relieve truck drivers from long hours of driving attention, and make logistic transportation more efficient and safer. In this project, a few donor trucks are equipped with additional systems to realize the platooning application. The added systems include a wireless V2V communication system, several computing units, and sensors such as proximity, localization sensors. Moreover, the platooning application requires access to various onboard sensors and actuators of the donor trucks. The platooning application enables the trucks to drive with a shorter inter-vehicle distance safely automatically. We show an impression of this project's results in Figure 7.3.

7.3.2 Safety Analysis with "Connected Vehicles" Approach

In the "connected vehicles" approach, next to the vehicle-centric HARA at the vehicle level, a SoS HARA is done to identify the hazards related to the coordination of the platoon, and the virtual vehicle represented by the platoon. We identified two categories of hazards: hazards for the other road users, and hazards for the trucks within the platoon.

In the "connected vehicles" approach, it is easier to visualize the role and associated hazard of the leading vehicle, that is the front interface of the system and the trailing vehicle, that is the tail of the system. The other road participants also interact with the vehicles in the platoon, for instance when they perform a cut-in/cut-through in between the platoon members. In this situation, the platoon disassembles into a set of single vehicles or smaller size platoons.

When performing the SoS hazard analysis, we investigate the hazards similar to the vehicle-centric approach too, but we translate those for a virtual vehicle that the platoon represents. Consequently, we identify hazards related to steering, braking, acceleration of the complete platoon, and the interaction with the driver, between the platoon drivers and with other road users. For example, when analyzing hazards related to braking, the emergency braking of the leading truck due to hazardous situations in front of the platoon, should be "communicated" fast enough to the other road participant behind the platoon. Since the platoon system also involves the V2V communication, all hazards and safety goals related to communication failures need to be analyzed at the "connected vehicles" level. Also, the "string stability" of the platoon can be investigated on the "connected vehicles" level taking into account the dynamics created due to each platoon members.

The FSC at the "connected vehicles" level focuses on the safety goals and safe states at that level. We project the functional safety requirements on the platoon members depending on the relative truck position within the platoon. Also, we address the requirements related to the V2V communication including the timing requirements at the "connected vehicles" level. We show an example of a fault tree resulting from the "connected vehicles" approach in Figure 7.4[1]. This tree shows the deductive analysis on the top event related to: "too small (less than 0.3s) inter-vehicle time gap".

7.3.3 Safety Analysis with Vehicle-Centric Approach

In the vehicle-centric safety analysis approach, the focus of the process is at the vehicle level. Therefore, hazards are defined based on which hazardous event can occur within one vehicle in the platoon. In this analysis and assessment each of the trucks, depending on their location in the platoon (leading, following, or trailing) is investigated. We identified the hazards that are similar to what is traditionally analyzed by Advanced Driving Assistance Systems (ADAS) and Autonomous Driving (AD) applications. At the vehicle level, these hazards are related to steering, braking, accelerating, and interaction with the driver (informing driver and receiving commands). These hazards focus more on the actuators of the vehicle that can change the behavior of the vehicle and result in possible collisions with other road participants (or with the trucks part of the platoon).

[1] The text in the analysis is mostly blurred out due to confidentiality issues

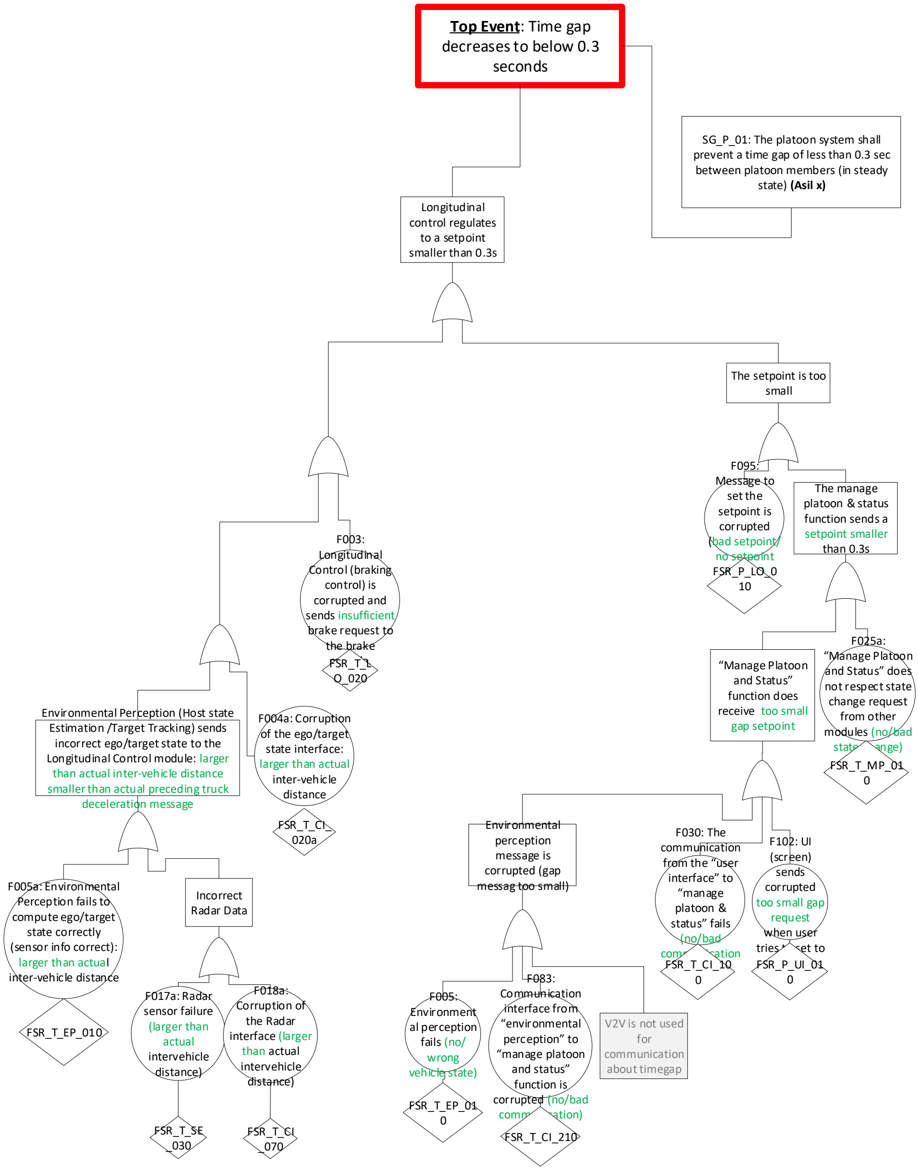

Top Event: Time gap decreases to below 0.3 seconds
SG_P_01: The platoon system shall prevent a time gap of less than 0.3 sec between platoon members (in steady state) (Asil x)
Longitudinal control regulates to a setpoint smaller than 0.3s
The setpoint is too small
F095: Message to set the setpoint is corrupted (bad setpoint/ no setpoint
FSR_P_LO_0 10
The manage platoon & status function sends a setpoint smaller than 0.3s
F003: Longitudinal Control (braking control) is corrupted and sends insufficient brake request to the brake
FSR_T_L O_020
"Manage Platoon and Status" function does receive too small gap setpoint
F025a: "Manage Platoon and Status" does not respect state change request from other modules (no/bad state change)
FSR_T_MP_01 0
Environmental Perception (Host state Estimation /Target Tracking) sends incorrect ego/target state to the Longitudinal Control module: larger than actual inter-vehicle distance smaller than actual preceding truck deceleration message
F004a: Corruption of the ego/target state interface: larger than actual inter-vehicle distance
FSR_T_CI_ 020a
Environmental perception message is corrupted (gap messag too small)
F030: The communication from the "user interface" to "manage platoon & status" fails (no/bad communication
FSR_T_CI_10 0
F102: UI (screen) sends corrupted too small gap request when user tries to set to
FSR_P_UI_01 0
F005a: Environmental Perception fails to compute ego/target state correctly (sensor info correct): larger than actual inter-vehicle distance
FSR_T_EP_010
Incorrect Radar Data
F017a: Radar sensor failure (larger than actual intervehicle distance)
FSR_T_SE _030
F018a: Corruption of the Radar interface (larger than actual intervehicle distance)
FSR_T_CI _070
F005: Environmental perception fails (no/ wrong vehicle state)
FSR_T_EP_01 0
F083: Communication interface from "environmental perception" to "manage platoon and status" function is corrupted (no/bad communication)
FSR_T_CI_210
V2V is not used for communication about timegap

Figure 7.4: An example of a fault tree resulting from the "connected vehicles" approach

In the FSC process, the main focus is on deriving functional requirements that cover the safety goals and focus mainly on the vehicle level functionalities. Therefore, we perform fault tree analyses to identify the platooning system failure modes that lead the violation of safety goal resulted from HARA. In this process, it is necessary to consider the effect of the failure modes for each relative position of the truck within the platoon, leading, following or trailing. We show, as an example, one of the resulting fault trees graphs in Figure 7.5; this graph is the result of analysis failure modes that may lead to the violation of the safety goal related to the prevention of insufficient braking.

Note that the fault labeled “Time gap too small ⋯” is similar (from the content perspective) to the top event from the example from the “connected vehicles” approach (See Figure 7.4). We were able to find this since the “time gap settings” is a parameter of the Longitudinal Controller in the platooning system design. However, it is only discovered in the fourth level of the tree here; whereas, previously, it was the top event of the tree. Further, note that the events under this particular failure are not the same as the ones under the graph in Figure 7.4. The nodes in this example reflect the functional architecture of the platooning system in the vehicle level, while the others reflect on the platoon level functionalities (which mostly reflect on the communication network).

7.3.4 System Level Development and Validation

The Technical Safety Concept (TSC) of the safety analysis process focuses on the mapping of the functional safety requirements on the hardware and software components (the deployment of functional and non-functional requirements). Since all safety requirements are eventually implemented within the platooning system of each truck, the functional safety requirements at vehicle and platoon levels are both mapped at the vehicle level systems and software.

Once the hardware and software development of the platooning system is concluded, the integration of the platooning system and later on at the vehicle level can be conducted similarly for both approaches. Because of the decomposition of FSRs at the truck and the platoon level in the “connected vehicles” approach, it is easier to identify the test cases that are supporting the safety of the truck, as a single entity, as opposed to the safety of the truck as being part of a platoon. This simplification and clarity of the “connected vehicles” approach increase the confidence in covering the verification and validation of all hazards mitigation.

7.4 Discussion

In this section, we proceed with the discussion on the results of the comparative study. There are two aspects that we pay attention in this discussion. First, we compare the capability of the two methods of providing complete and correct safety analyses. Second, we consider the efficiency of these methods by comparing the necessary effort for each approach.

The traditional vehicle-centric approach faces a few challenges on safety consideration during concept development for a SoS. To start with, expressing the requirements for the safe states is complicated when using the vehicle-centric approach. As an example, Figure 7.6 shows the critical path for emergency braking in the platoon system. This example shows the dependency of the emergency brake decision of the following truck on the sensory inputs of the leading truck. Any failure in the elements of the leading truck in the critical path should trigger a preemptive reaction of the following truck and a reconfiguration of the controller to a degraded mode. Considering the platoon system in the safety analysis, it is easy to infer (as shows with above example) that a critical failure (such as braking system failure) in a platoon member (e.g., the lead truck) should require activation of a safe state in that member as well as a degraded operating mode for the other members (e.g., the following truck). However, considering only the truck system in the safety

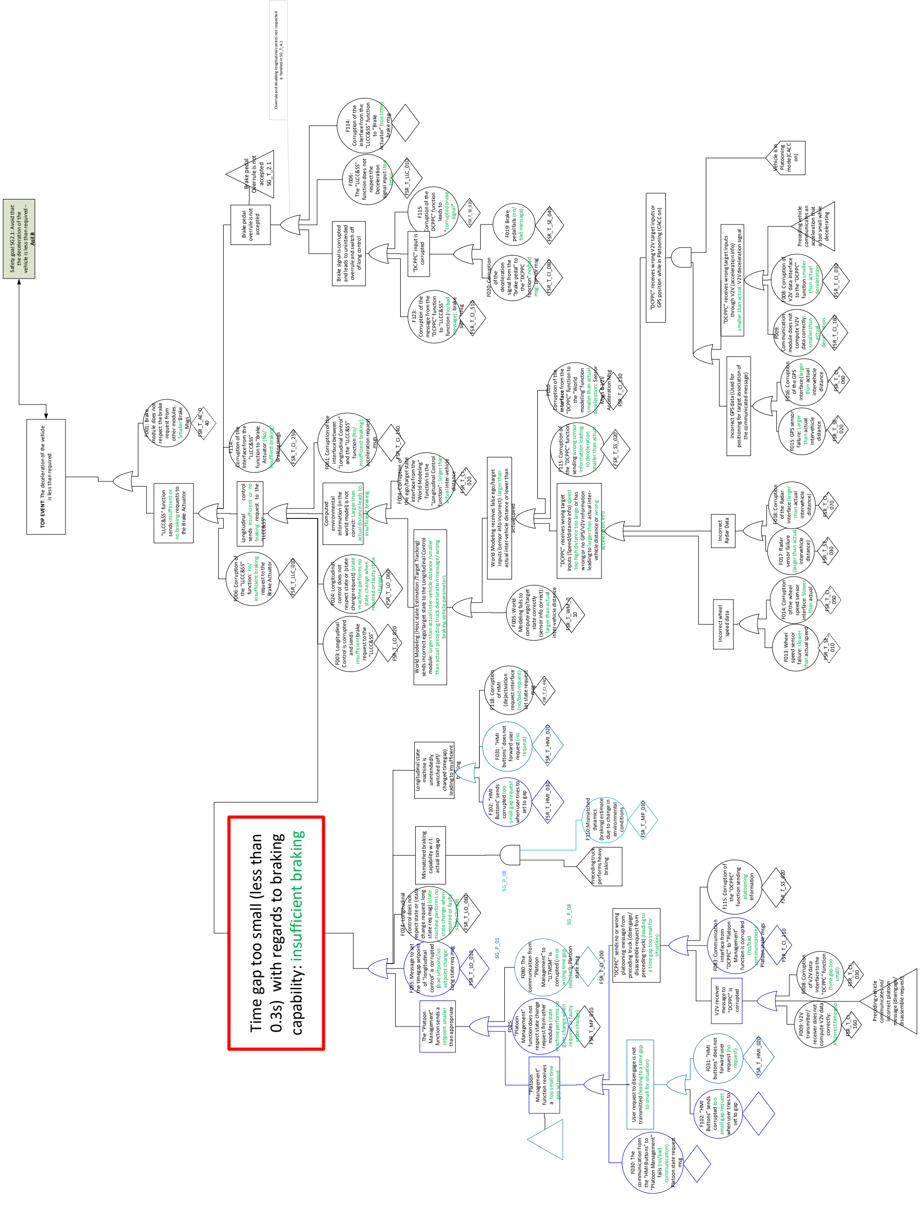

Time gap too small (less than 0.3s) with regards to braking capability: insufficient braking
TOP EVENT: The deceleration of the vehicle is less than required
Safety goal SG2.1: Avoid that the deceleration of the vehicle is less than required – ASIL B
"LLC&SS" function sends insufficient or no braking requests to the Brake Actuator
Brake pedal overrules not accepted
Brake signals corrupted and leads to unintended overrule and switch off of long control
World Modeling fails to compute ego/target state correctly
Incorrect GPS data (used for positioning for target association of the communicated message)
Incorrect Radar Data
Incorrect wheel speed data
Mismatched braking capability w.r.t. actual timegap
Longitudinal state machine is unintendedly switched (off/change timegap) leading to insufficient braking
The "Platoon Management" function sends a setpoint smaller than appropriate
User request to disengage is not transmitted (leading to a time gap too small for situation)
Vehicle is in Platooning mode (CACC on)
Preceding truck performs heavy braking
SG_P_01
SG_P_08
SG_P_08
SG_T_2.1

Figure 7.5: An example of a fault tree resulting from the Vehicle-Centric approach

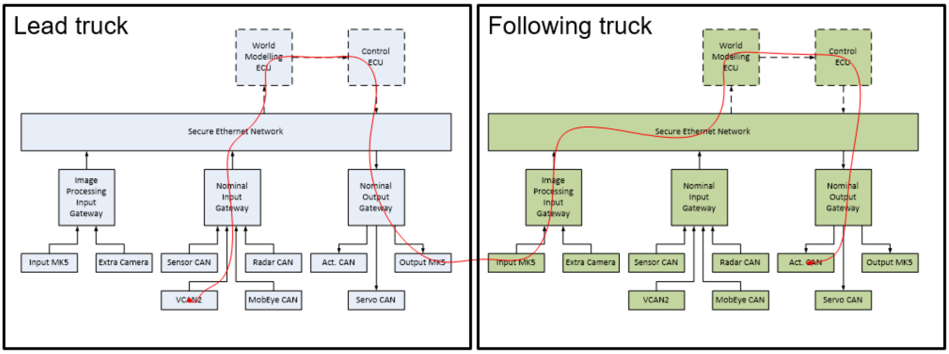


Figure 7.6: The critical path of emergency braking in a platoon system

concept development makes it relatively difficult to capture the requirement on the other platoon members.

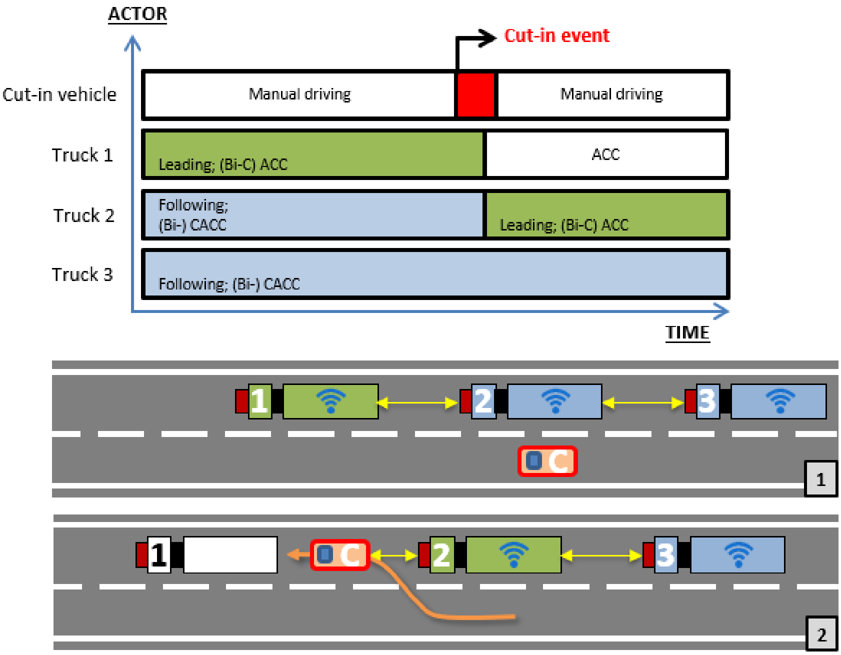


Figure 7.7: An illustration of a cut-in scenario as an example of a "connected vehicle" scenario.

Another challenge is capturing all the possible situations that a SoS faces in the vehicle-centric approach. In Figure 7.7 we give an example of a scenario that is only sound for a platoon system: some other road user performer cut-in maneuver in front of the middle following tuck of a platoon as shown in Figure 7.7. The platoon in this scenario is broken after the cut-in event, the previous platoon leader drives on Adaptive Cruise Control (ACC),

Table 7.1: Fault Tree Analyses Statistics

Approach	# Safety Goals	Total # Nodes	Mean # Nodes	# SVN Commits
“Connected Vehicles”	23	403	17.5	39
Vehicle-centric	11	353	32.1	44

while the middle following truck has to pick up the leading task of the platoon. In this scenario, the cut-in event is *sensed* by the second truck, and this event resulted in the change of behavior in the first truck. If the platoon system were not considered in this scenario, then the situation would change to an overtaking maneuver with no impact on the first truck. This could potentially lead to inadequate hazard identification in the vehicle-centric approach. Simply because with this approach not all behaviors can easily be described.

The results of the comparative study showed that both approaches are capable of covering all safety-related issues. We exemplified this capability using the two fault tree graphs. However, we showed that the vehicle-centric approach is more prone to error due to complexity.

On the other aspect of our comparison, we pay attention to the effort required for each approach. Table 7.1 gives an overview of the statistics of the safety analysis performed during each approach. This table gives the total number of safety goals determined, the total number of nodes in the fault tree analyses, the average number of nodes in the fault tree analyses, and the number of total revision commit (only considering the safety concept work) for each approach.

The additional safety goals in the “connected vehicles” approach, are the result of the fact that the platooning system has more functionalities when the item is defined considering using the “connected vehicles” approach. Because the functions are defined differently, the hazard analysis results in different hazards and therefore different safety goals are formulated. Although there are fewer safety goals in the vehicle-centric approach, the average number of nodes in the fault tree analyses is almost twice as high in the “connected vehicles” approach. We infer that the vehicle-centric approach results in higher complexity in the safety analyses. An increase in complexity means that the safety experts needed more effort for delivering results, which is confirmed by the number of revisions.

7.5 Conclusions

The primary motivation of the “connected vehicles” approach, presented in this chapter, is the need to establish a safety case that acknowledges that failure modes within a truck have varying effects and consequences whether it occurs while the truck is driven on its own or as part of a platoon. Therefore, the set of hazards and associated safety mechanisms at the truck level are different from the ones at the platoon level. This variation implies that the two main statements for a platoon system’s safety case are: First, each vehicle of the platoon is intrinsically safe; and second, the platoon is safe both as a formation of vehicles and as a virtual vehicle to other road users. These two statements entail that the safety of the trucks participating in a platoon is ensured as well as the safety of the other road users encountering or interacting (e.g., cut-in) with the platoon. We believe that our proposed method increased the trust in the safety of the platooning system by breaking down the complexity of the analyses and providing more transparent tractability of risk mitigation mechanisms at various abstraction levels.

In an ecosystem of connected automated vehicles, several other elements of the ecosystem (infrastructure information, coordination center) and their relative contribution to the overall ecosystem's safety or the safety of any element of the ecosystem needs to be considered. Although, the scope of this chapter and the related project (EcoTwin) is limited to a simple ecosystem composed only of the trucks within the platoon; we expect, nevertheless, that the approach presented here can easily be expanded to a more complex ecosystem. This expansion may be required adding additional levels of abstraction and duplicating each relevant step of the proposed functional safety process (Item definition, HARA, FSC, V&V at the abstraction level) accordingly. Lastly, we did not consider cyber-security in this research; however, this quality aspect would need to be addressed in a similar way such as to demonstrate the security of each element of the eco-system as well as the inherent security of the whole eco-system.

Chapter 8

A method for measuring safety culture based on ISO 26262

Safety culture is the collective attitude of members of an organization regarding safety issues such as awareness, communication, and knowledge. In the automotive industry, specifically in its Research and Development (R&D) environments, safety culture is relatively new. Recent incidents related to functional safety issues in automotive software and hardware, call for the improvement of safety culture in R&D environments. In ISO 26262 safety culture is identified as a requirement for safety management. Improving on safety culture is essential for using this standard as a way of working. In this chapter, we introduce a method for measuring the safety culture per ISO 26262. We quantified safety culture based on participants' response to a questionnaire. We measure several contributing factors such as management commitment, awareness, the flow of information, knowledge and skills. For each contributing factor, a set of survey items are designed and verified by external experts' reviews. We selected the final questions from the survey pool based on experts' feedback. Finally, we performed the survey at the Department of Integrated Vehicle Safety (IVS) of TNO, an R&D environment. We obtained an indication of the current status of safety culture; furthermore, the survey provided new insights into improvement points for the IVS Department.

This chapter is based on:

[62] A. Khabbaz Saberi, F. Benders, R. Koch, J. J. Lukkien, and M. G. J. van den Brand, "A method for quantitative measurement of safety culture based on ISO 26262," *Evolution of System Safety: Proceedings of the Twenty-Sixth Safety-Critical Systems Symposium, 6-8 February 2018*, York, United Kingdom, p. 203-218, 2018

8.1 Introduction

Since 1986 when a poor safety culture was indicated as a root cause of the Chernobyl disaster, the concept of safety culture has been the focus of research in several safety-critical domains. Industries such as avionic, health, railway, and energy have a history of safety culture [39], [41].

The ISO 26262 standard [52], the functional safety standard in the automotive domain, indicates safety culture as one of the requirements of overall safety management. In general, it requires the organization to provide a proper environment for people involved in safety-related activities. The automotive industry is taking on a new challenge with the advent of automated driving. Introducing true automated driving to the market requires the addition of many features to today's vehicles. This increase of complexity brings new challenges for ensuring safety in the automotive industry. The effort that this industry is putting into defining the safety of the intended functionality in ISO/PAS 21448 [53] is evidence of the importance of this new challenge.

With the competition in this industry and the focus on decreasing the time-to-market for releasing new features, the gap between development and production is smaller than ever. As addressed in ISO/PAS 21448, many of the safety issues need to be addressed during the advanced development phase.

Considering the great impact of early design decisions during the research and development phase on safety magnifies the importance of safety culture during the early development phases. If safety is not responsibly considered during the development phase, there is a possibility that design decisions and trade-offs negatively impact safety. Therefore, it is essential to have a mature safety culture in the development team or organization to ensure that safety-related issues are identified and tackled properly.

This necessity raises a question on how to assess the maturity of the safety culture of an organization. Answering this question is essential since traditionally the automotive Research and Development (R&D) is not accustomed to safety engineering. Here we consider R&D to be any organization or team that is in charge of the advance development.

The safety culture is defined by [101] as follows:

> "The set of enduring values and attitudes regarding safety issues, shared by every member of every level of an organization. The Safety Culture refers to the extent to which every individual and every group of the organization is aware of the risks and unknown hazards induced by its activities; is continuously behaving so as to preserve and enhance safety; is willing and able to adapt itself when facing safety issues; is willing to communicate safety issues; and consistently evaluates safety related behavior."

Some literature, especially those with an organizational psychology viewpoint such as [16], [34], differentiate between the safety culture and "safety climate." They argue that most efforts for measuring the safety culture measure the safety climate, as the latter is the superficial aspect of the safety culture. In this study, however, we have a more pragmatic viewpoint. Therefore we ignore the difference between the safety culture and climate. Making a distinction between the two is not helpful for us; since we focus on methods of measuring safety culture and do not intend to study the psychological aspects of it.

A qualitative method for measuring the safety culture is proposed by [15]. They conclude that surveying the safety culture can positively influence it in that organization since it stimulates a discussion on the matter. Furthermore, a method for a quantitative assessment of safety culture is proposed by [131]. In this method, the key aspects of safety culture are identified and assessed during interviews conducted using both closed and open questions.

Interesting research was done by [37] in the Avionic domain. They conducted a questionnaire based survey in an avionics R&D environment; and based on the survey's findings, suggested improvement points on the safety culture of the target organization. The difference between this research and our survey (other than the domain) is the attention to norms and standards in the respective domains concerning functional safety.

In this chapter, we describe the design of a questionnaire with the focus on the safety culture for automotive R&D. The contribution in this work is first, to focus on the topic of the safety culture in the context of functional safety in this domain. Second, we design a questionnaire for assessing the safety culture; and apply it for assessing the safety culture in a research organization.

The rest of this chapter is organized as follows: In Section 8.2, the methodology and the design of the questionnaire are described. The case study and the results are described in Section 8.3. In Section 8.4 validation of this research is presented. Finally, Section 8.5 concludes this chapter.

8.2 Survey Methodology and Description

The survey in this research follows the process described by [32]. There are several phases in this survey: Study Definition, Design, Implementation, Execution, Analysis, and Packaging.

In the Study Definition Phase, the survey goals and research questions are defined, as well as the context of the survey. The Design Phase covers converting the study goals into questions and addresses the validity threats. In the next phase (Implementation) the survey items are put in an executable format; moreover, the survey items are fine-tuned based on results from external reviews by field experts and a pilot survey. The Execution Phase, as the name suggests, is dedicated to performing the actual survey to collect data. In the Analysis, and Packaging Phase the results are interpreted and put into a useful format.

In this section, we describe the survey according to the phases as mentioned above. We describe the information related to the Study Definition, Survey Design, and Implementation phases here, and discuss the rest in Section 8.3.

8.2.1 Study Definition

We define the objective of this survey in the following survey goal:

Survey Goal: *Measuring the safety culture per ISO 26262 in a research and development organization.*

There is currently less safety culture in an R&D environment compared to the production environment in the automotive domain. This state needs to change due to the high impact of R&D on the developed functionality of modern vehicles, especially considering the current trend on reducing time to market in this industry.

What makes the safety culture of an R&D environment different from a production environment, is the importance of knowledge and skills of people in the systematic approach towards safety-related systems development. Using an accepted safety standard could be

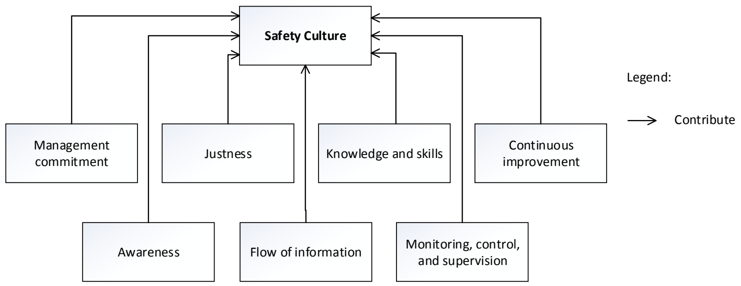


Figure 8.1: The contributing factors of the safety culture

helpful in addressing this aspect of safety culture. Although ISO 26262 is mostly concerned with the development and production of automotive systems, it is still applicable to an R&D environment since this standard addresses both product and process aspects of safety. Besides, the addition of ISO/ PAS 21448 will address advanced development soon. In this subsection, we elaborate the stated survey goal and describe its elements in detail.

8.2.1.1 Safety Culture

Based on the safety culture definition as given in Section 8.1, an overview of the contributing factors to the safety culture is shown in Figure 8.1. These key factors are the result of the aggregation of the aspects considered in the literature [101], [131]. The description of these factors adapted to the context of this research are as follows:

Management commitment is the willingness of the organization at every level (from top to down) to invest and prioritize effort in safety and their genuine positive attitude towards safety. The ISO 26262 standard emphasizes this factor in the clause Part 2 5.4.2.1, and 2.

Justness (only considered in [101]) is the extent to which behavior according to functional safety is encouraged and rewarded by the organization. Moreover, there should be a "no blame" culture where, in event of an accident solutions are sought instead of blaming the responsible person. The ISO 26262 also mentions this matter in the clause Part 2 5.4.2.1.

Awareness is the level of individuals' appreciation of their role and impact on functional safety in this context, and on safety in general. Moreover, their understanding of the risks involved in their work for themselves and others is also part of awareness. The ISO 26262 standard addresses the issue of roles in the clause Part 2 5.4.2.2.

Flow of information is the accessibility of new information for the right people through transparent communication. For instance, if there is a new hazardous situation identified during a recent test, the information should be easily provided to others, to be considered if applicable to their projects. In ISO 26262 Part 2-5.4.2.3 the flow of information is mentioned as explicit communication of functional safety anomalies. The ISO 26262 standard even takes the flow of information further by stating that there shall be a process for resolving functional safety anomalies in the clause Part 2-5.4.2.4.

Knowledge and skills (similar to "behavior" in [101]) are the extents of individuals' knowledge of safety engineering processes and activities, and in particular in this case, the ISO 26262 standard. This factor is of more importance in an R&D environment in comparison to a production environment. General appreciation of the relevant knowledge

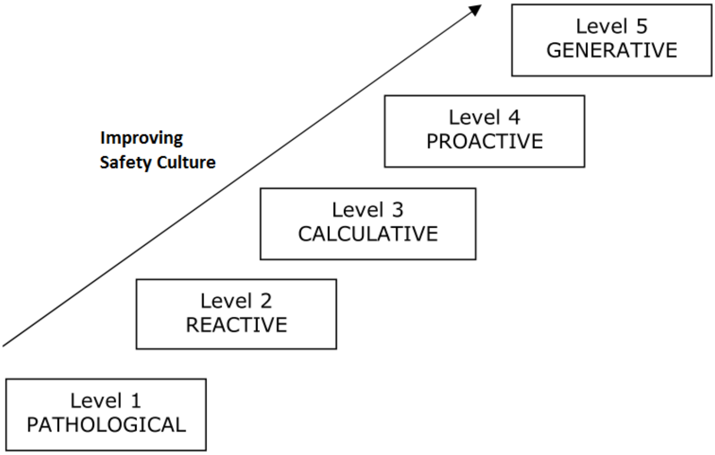


Figure 8.2: Safety Culture Maturity Model [101]

and skills are needed in the organization to allow effective implementation of functional safety. Several clauses of ISO 26262 can be linked to this aspect of safety culture such as Part 2 5.4.2.5, and 6.

Continuous improvement (the same as “adaptability” in [101]) is the willingness of the organization to learn from their experience and improve the way of working of the organization. Continuous improvement is mentioned in clause Part 2 5.4.2.7 of ISO 26262.

Monitoring and control (only considered in [131]) are the existence of supervision mechanisms concerned with safety and the visibility of these mechanisms in the organization. Moreover, the extent to which the required authority is provided to execute functional safety is also part of this aspect. The supervision issue can be traced in ISO 26262 in the clause Part 2 5.4.2.8.

8.2.1.2 Safety Culture Metrics

A model for the safety culture maturity is introduced by [46]. Similar work was done by [30] on safety culture maturity model. An overview of the maturity model is shown in Figure 8.2. Similar to the Capability Maturity Model (CMM), the safety culture maturity model has five levels. The general idea is that an increase in the level shows improved safety culture maturity.

The first level, indicating the worst safety culture, is when an organization sees safety as a burden. There are typically no processes in place for dealing with safety issues, and people only care about not getting into trouble. The second level is applicable when there are some processes for safety but not strictly followed by staff. It could be that the management of the organization states that safety is important, but the members do not believe it. In the next level, i.e., the calculative level, the safety processes are followed, and the members are more involved in the safety issues. Nevertheless, the safety processes are not believed to be critical. In the proactive level, both management and staff believe in their safety processes, and all hazards are addressed systematically. In the last level, safety is an organization value. Both members and management are constantly improving safety. More details can be found in [44].

Table 8.1: Mapping safety culture metric to the safety culture maturity level

Safety culture maturity level	Pathological	Reactive	Calculative	Proactive	Generative
SC score	[0 0.5]	(0.5 0.65]	(0.65 0.8]	(0.8 0.9]	(0.9 1]

To be able to assign one of the mentioned levels to an organization, it is required to process the results from a survey and map the outcome to one of the levels. Here we give a detailed processing method on survey results and suggest a possible mapping.

Taking the average of the participants' scores is commonly used to measure safety culture based on a survey. Each participant is scored by taking the normalized weighted average of their response. This score, referred to as the Individual Score (IS), reflects on how the participants view the organization concerning safety issues. This method is also used in work of [131].

The metric for safety culture of individuals is proposed by means of a normalized weighted average as in Equation 8.1:

$$IS = \frac{1}{2} + \frac{\sum_{i=1}^{n} w_{p,i} \cdot s_i \cdot q_i}{4\sum_{i=1}^{n} w_{p,i}} \tag{8.1}$$

where n denotes the number of survey items; s_i is the sign of the desired answer ($+1$ for agreeable questions, and -1 for disagreeable questions); and q_i denotes the answer to each question. We assume that q_i is in range of $[-2\ 2]$. Moreover, $w_{p,i}$ is the weight of each question for that participant. The weights enable the score to be adjusted for each participant based on their role in the organization and type of work they do.

The weighted average is normalized such that the outcome becomes a real number in the range of [0 1] with zero indicating the worst possible and one the best possible outcome.

We use the individual safety culture metric to define safety culture for the organization as the mean of individual awareness over all the sample population:

$$SC = \frac{\sum_{i=1}^{l} IS_i}{l} \tag{8.2}$$

where l is the number of sampled questionnaires.

We propose a preliminary suggestion for mapping the safety culture metric to the safety culture maturity levels in Table 8.1. We take this mapping as an initial educated estimation. To arrive at an acceptable mapping, we require gathering more data by performing this survey in different organizations with different safety culture and comparing the resulting scores with experts' judgment. Nevertheless, for this particular mapping, a few factors were considered. An indifferent population sample scores 0.5 by always choosing the middle option. Assuming that this sample should be ranked at the pathological level, we map the first category to the [0 0.5] range. Another hypothetical population sample who answers all the survey items with moderate agreeable answers scores 0.75. Since this sample does not show strong opinions, we assume the calculative level for it.

8.2.2 Survey Design

Following our survey methodology, in this subsection we describe the Survey Design. We considered two options for the design of this survey: interview and questionnaire. Each option has some variations concerning question types. To be able to target a large population

Table 8.2: Sample of survey items

Contributing factor	Expected answer	Survey item
Awareness	+	I am aware of the main risks in my projects with respect to functional safety.
Awareness	+	I think that functional safety is vital for our business.
Awareness	+	I feel responsible for functional safety of my models/designs/ products.
Monitoring and control	+	I know that there are safety audits carried out for safety critical projects.
Monitoring and control	+	In my projects, review processes for functional safety are performed.
Monitoring and control	+	I know that there are safety control procedures within our department.
Flow of information	+	I get informed if there is a functional safety anomaly in my projects.
Flow of information	+	I know whom to inform about functional safety irregularities in my projects.
Flow of information	+	I know the procedures to follow when I find a functional safety anomaly.
Continuous improvement	+	There is a safety team responsible for improving functional safety.
Continuous improvement	+	I think that my colleagues do all they can to improve functional safety.
Continuous improvement	+	After an accident or near miss, we take actions to reduce the chance of it happening again.
Justness	-	I think that I will be blamed if there is a mistake in my work.
Justness	+	I think that I will be rewarded if I act to improve functional safety.
Justness	+	I think that all colleagues are able to express their concerns with respect to functional safety issues.
Knowledge and skills	+	In my projects, safety activities are distinguished in the project plan.
Knowledge and skills	+	Emergency operation is the functionality of transition to safe state.
Knowledge and skills	+	Safety analysis such as FMEA, and FTA help avoiding systematic failure.
Management commitment	+	I think that management does everything in their power for improving functional safety.
Management commitment	+	I think that functional safety is currently an important value in the policy of [Name of organization].
Management commitment	-	I think that in my projects time and cost have priority over safety.

(50 to 100 participants), and to minimize the participation time (targeted to be less than 20 minutes), we chose a questionnaire with structured, closed questions.

The advantage of a questionnaire (in comparison with an interview-based survey) is the high coverage of an organization at a relatively low cost. Furthermore, a questionnaire is more anonymous which assures that the organization members answer the questions freely without facing the consequences for having criticism on the organization. This anonymity is especially helpful in organizations that have a less mature safety culture. On the other hand, the drawback is that the results of the survey may be subject to various errors such as “socially desirable” answers, neutral answers, or substantive answers while participant does not know the answer.

There are a few validity threats associated with the chosen survey format such as choosing for “socially desirable” answers, neutral answers, substantive answers while participant really does not know the answer.

We surveyed the organization anonymously to minimize these validity threats regarding the social desirability. The possible responses range from “strongly agree,” to “strongly disagree.” The expected answers can either be agreement or disagreement. Moreover, a “do not know” option is available to provide a neutral answer as well as to give an option for those participants who genuinely do not have an opinion about a survey item.

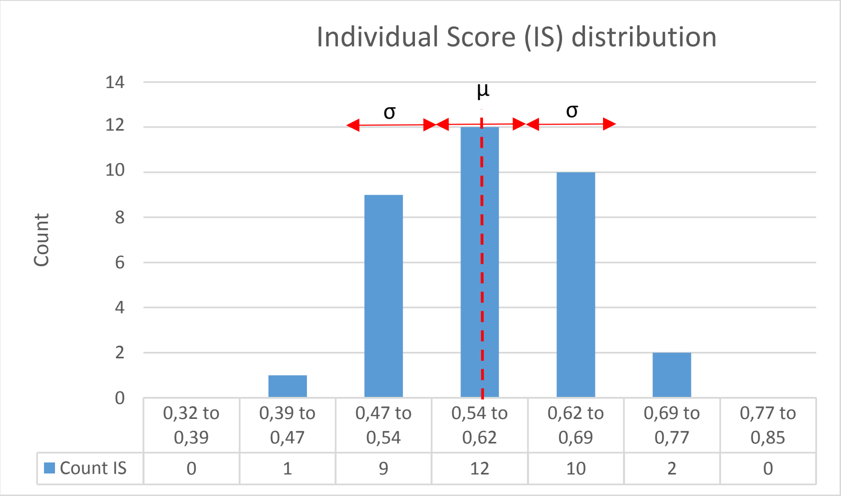
We used two sources to derive the survey items. We derived the survey items related to knowledge and skills from the industry standard of safety. The ISO 26262 standard is used as a reference to measure the knowledge and skills aspect of safety culture. The knowledge is measured based on the participants’ understanding of a selected vocabulary, and their familiarity with key concepts of ISO 26262. The key concepts chosen for this survey are as follows: safety lifecycle and safety plan, item and item definition, Hazard Analysis and Risk Assessment (HARA), Automotive Safety Integrity Level (ASIL) and ASIL decomposition, safety concept and safety analysis (FMEA, FTA, etc.), verification and validation, safety case.

The second source for the survey items is based on the literature. The survey items related to other contributing factors (other than knowledge and skills) are inspired by checklists and questionnaires in the literature [35], [101], [131]. We only adapted questions to apply to the specific situation of the domain and organization. A few sample questions are shown in Table 8.2.

8.2.3 Survey Implementation

Based on the selected vocabulary and key concepts of ISO 26262 a pool of survey items was designed using brainstorming methods. These questions, as well as the selected questions from the literature, were reviewed by safety experts. Next, based on the safety expert’s review feedback, several questions were selected to be used in the survey. We selected a total number of 71 questions out of 133 questions from the initial survey items after the review. Next, the questions were tested using a survey pilot among a selected team. After the pilot survey, the questions were refined to reduce the ambiguities found in the pilot and review. Finally, the questionnaire was put in paper and pencil format.

The questionnaire consists of two parts. The first part of the questionnaire is dedicated to the classification of the participant in the organization structure. These are questions about participant role in the organization, experience. This part needs to be tailored based on the structure of the organization and the chosen scope for the survey. The second part



	0,32 to 0,39	0,39 to 0,47	0,47 to 0,54	0,54 to 0,62	0,62 to 0,69	0,69 to 0,77	0,77 to 0,85
🟦 Count IS	0	1	9	12	10	2	0

Figure 8.3: The distribution of obtained Individual Safety Culture (ISC) scores

includes the survey items, which evaluate the individual safety culture based on the contributing factors.

8.3 Case Study

The questionnaire was used to assess the safety culture at the Integrated Vehicle Safety department (IVS) of TNO[1]. This survey aimed to establish a baseline on the safety culture maturity level of IVS. The IVS department has a flat organization with only a few defined roles: research manager, project manager, and researcher.

8.3.1 Execution

The survey targeted all the employees of IVS (both management, and researchers). There were more than 64 questionnaires distributed, and 34 were collected in one week.

8.3.2 Results Analysis

The distribution of participants individual score from the survey is shown in Figure 8.3. The graph is made by counting the number of ISs falling in intervals determined by the standard deviation. As the graph shows, the individual scores have a normal distribution with a mean value of 0.58 and a standard deviation of 0.08. Using a confidence level of 95%, the confidence interval of the safety culture is $[0.55\ 0.61]$. This shows that the safety culture is in the level 2 with a confidence level of 95%.

An overview of the contributing factors to safety culture is shown in Figure 8.4. The box-plot graph shows the distribution and mean value of the scores per each contributing

[1] TNO is an independent research institute with the mission of connecting people and knowledge and creating innovations that boost the sustainable competitive strength of industry and well-being of society [126]. The Integrated Vehicle Safety (IVS) department of TNO focuses on development of automated driving technologies.

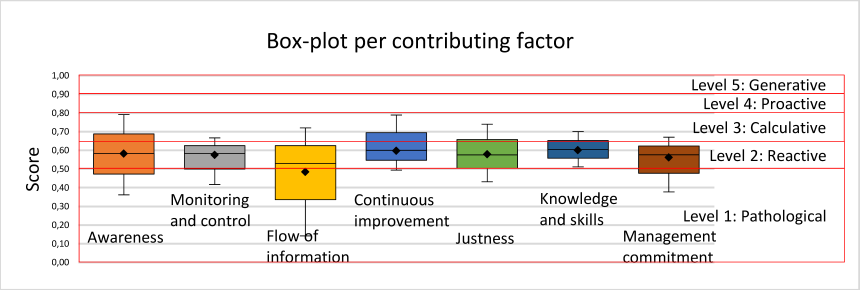


Figure 8.4: The box plot showing the results scores of contributing factors

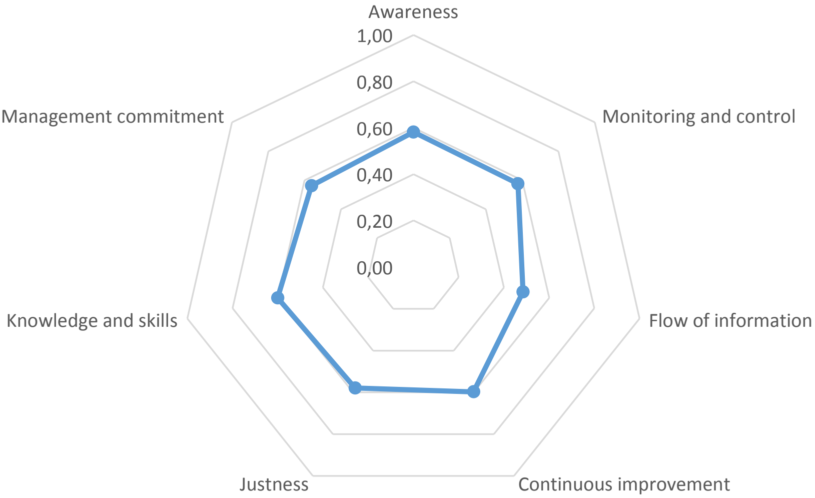


Figure 8.5: The spider web graph comparing the scores of contributing factors

factor. Moreover, the red boxes show the five levels of the safety culture maturity model. It can be seen that except for the flow of information, all the contributing factors fall into the level 2 of the safety culture maturity model.

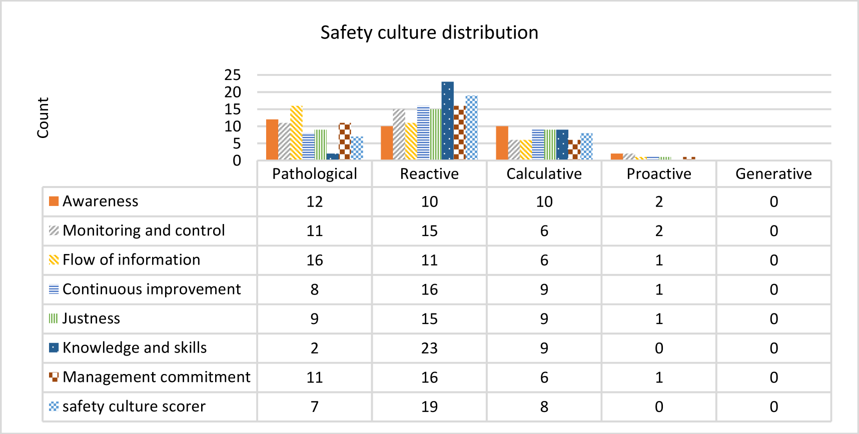
A spider-web graph of the contributing factors is shown in Figure 8.5. In this graph, it can be easily seen that the flow of information is the contributing factor with the lowest score.

A bar chart of the distribution of the contributing factors, as well as the SC is shown in Figure 8.6. This graph shows the number of people in each level of the safety culture maturity model with respect to each contributing factor.

8.3.3 Discussion

The results show that the safety culture at IVS is at the reactive level. Moreover, based on the results, the flow of information could be an improvement point at IVS.

As described, we give a detailed overview of the results of the scores of the IVS department in Figure 6. The figure shows that most of the individual scores are in the reactive



	Pathological	Reactive	Calculative	Proactive	Generative
Awareness	12	10	10	2	0
Monitoring and control	11	15	6	2	0
Flow of information	16	11	6	1	0
Continuous improvement	8	16	9	1	0
Justness	9	15	9	1	0
Knowledge and skills	2	23	9	0	0
Management commitment	11	16	6	1	0
safety culture scorer	7	19	8	0	0

Figure 8.6: The bar chart showing the distribution of the safety culture

level (level 2). Most of the employees are aware of the functional safety guidelines but react to hazardous situations that occur. It can be noted that the information related to safety is not flowing well and that only those who are directly working on functional safety are aware of how the information should flow.

To improve the safety culture, IVS researchers should be notified that there is a process to improve the quality of the safety culture and they can report issues related to safety. The knowledge and skill will grow through workshops and presentations are organized to increase the awareness and usage of the functional safety processes. The IVS management is committed to improving the safety culture since safety is one the most important enablers to develop prototype systems that can be tested on public roads.

8.4 Validation

We validated this research in two steps: First step is the internal validation that cares about the results of the survey. Internal safety experts perform this validation. The second one is the external validation that considers the validity of the process of this research, as well as the correctness of the questionnaire designed.

8.4.1 Internal validation

In 2015 the IVS department started to implement and integrate the functional safety based on ISO 26262. The survey is used to assess the current maturity of the department on integrating the safety culture in the way of working. Since the process started just a year prior to the survey, the maturity in the safety culture has been growing. The reached level (level 2: reactive) is according to the expectations considering the nature of the work at the department: i.e., research in the development and assessment of prototype automated driving systems. Also, considering the multidisciplinary composition of the expertise at IVS, the current way of working, the diversity of projects that are performed, and the effort spent in integrating ISO 26262 in the department, the results were no surprise for IVS.

The results showed that especially the large growth of the department with new young employees gives a large spread in the aspect of the information flow and awareness. The survey was particularly helpful to identify the current status/level and what should be improved to move to a more mature level. Furthermore, it also shows the focus areas that should be selected to improve the safety culture.

8.4.2 External validation

The method for quantitative measurement of safety culture as well as the outcome of a sample application of it as described above have been reviewed by Ricardo[2] as external validation. Though limited in both depth and size of the questionnaire, (to keep the method practically applicable) the method was found to be valuable for a quick survey on the actual safety culture within middle sized to large groups (typically $[50-250]$ people).

It generates a value which can be used as an indicator for comparing the safety culture between different groups or comparing the safety culture within one group developing over the years.

8.5 Conclusions

In this chapter, the design of a survey for quantifying the safety culture in an automotive research and development environment was presented. The survey was performed at the IVS department of TNO. Internal and external safety experts validated the results.

According to the survey, IVS has safety maturity of level 2. Furthermore, the lack of information flow was identified as the bottleneck in the safety culture. The results gave some insight on how to improve the safety culture at IVS.

The case study conducted at IVS showed potential for this questionnaire to be an indicator of safety culture maturity level. The questionnaire provides a low-cost solution to gain insight into the safety culture of an organization, as well as identifying the points for improvement. Although, more surveys are needed to validate the suggested mapping of SC to the maturity levels.

As future work, we intend to improve the questions based on received feedbacks and repeat the survey at IVS to track the possible improvements in safety culture. Moreover, the mapping of the safety culture score based on the questionnaire to the safety culture maturity model could be improved based on these surveys.

[2] Ricardo plc is a consultancy firm active throughout the world in the fields of strategy, technology, environment and safety. Ricardo is also a specialist niche manufacturer of high-performance products. The firm has in excess of 2,900 expert engineers, consultants and scientists working in its core areas of engines, gearboxes, vehicles, hybrid and electrical systems, and environmental forecasting and impact analyses. Ricardo plc is active in a wide array of market sectors, including automotive, rail, defense and energy. Ricardo Nederland is situated in Utrecht and employs over 200 technical specialists.

Chapter 9

Conclusions

In this chapter, we summarize the contributions of this thesis and reflect on the research question. We also indicate some directions for future work.

9.1 Contributions

Throughout this thesis, we discussed various topics related to the integration of functional safety with system design in the automotive domain. In particular, we looked into the ISO 26262 and its requirements for system development as discussed in Chapter 2. Our first research question was regarding the integration of the process and system design aspects of safety:

> **RQ 1:** *How can domain models of functional safety cover both system design and process aspects?*

We discussed this question in Chapter 3, where we introduced the Holistic Safety Domain Model (HSDM). We based this model on a systematic analysis of the specification of the ISO 26262. We analyzed and modeled the conceptual development (Part 3) of this standard. Our proposed domain model formalizes both the system design and process aspects. We further show two applications of our proposed model: modeling the workflows of the safety lifecycle and modeling safety analysis specifications.

We showed that modeling the process aspect was possible and that it can play a role in compliance checking, which leads to our second research question:

> **RQ 2:** *How can model-based techniques be used for compliance assurance?*

We addressed **RQ 2** in Chapter 4. We define constraints based on ISO 26262 and a domain model of this standard. We also provide a software tool to show the feasibility of our proposed method. We designed and implemented a software tool to interface with two other software tools, i.e., Microsoft Excel and Enterprise Architect (EA), and to keep a common project model. Our tool evaluates the defined constraints on the project model. Our tool checks the constraints for the failure of compliance with the ISO 26262. Based on this evaluation, the tool provides feedback to show certain design mistakes are made. Our proposed method makes it possible to provide feedback in a shorter time as opposed to checking the models at the end of the development. Timely feedback may reduce the impact of possible noncompliance and human errors. By automating specific tasks and detecting noncompliance at an early stage, we make it possible to reduce the overall development time and compliance evaluation effort.

Our next research question addresses the more technical issues of safety-critical system design:

> **RQ 3:** *How can architectural patterns be used for achieving functional safety in automated driving applications?*

We addressed this question in Chapters 5 and 6. We presented a novel architecture pattern (Safety Channel) suitable for the architecture of automated driving applications. We further demonstrate a generic approach to compare the proposed pattern with some selected patterns, according to several quality attributes. We use an extended version of Safety Channel to design the functional safety architecture of two automated driving applications.

Furthermore, we discussed the system of systems aspect of functional safety by **RQ 4**:

> **RQ 4:** *What is the impact of system of systems composition on safety analysis?*

In Chapter 7 we proposed the "connected vehicles" approach. We show that the overall safety of an SoS requires statements both on the vehicle level as well as the SoS level. We show the applicability of our method with a platooning application where multiple trucks use a V2V communication network for distributed control. We believe that our proposed method increased the trust in the safety of the platooning system by breaking down the complexity of the analyses and providing more transparent tractability of risk mitigation mechanisms at various abstraction levels.

Finally, we address the organizational aspect of functional safety in **RQ 5**:

> **RQ 5:** *How to measure the safety culture in advanced development or research organizations?*

In Chapter 8 we address this question. We designed a survey for quantifying the safety culture in an automotive research and development environment. We performed the survey at the IVS department of TNO and validated the results via internal and external safety experts. The case study conducted at IVS showed potential for this questionnaire to be an indicator of safety culture maturity level. The questionnaire provides a low-cost solution to gain insight into the safety culture of an organization, as well as identifying the points for improvement. Although more surveys are needed to validate the suggested mapping of SC to the maturity levels.

To conclude, we have investigated methods for improving the integration of functional safety in automotive systems design. We refined this primary research objective into five more refined research questions to study the various aspects of the integration. We answered these questions in six chapters of this thesis. The findings of this thesis have been applied in several projects of IVS, namely: EcoTwins (I, II, and III), i-Combi, and TULIP[1] that are oriented around the truck platooning application; as well as European funded projects such as ASSUME [103] and ROADART [108]. Most notably, the TULIP project (currently under development) uses the integrated V-model as the development process (discussed in Chapter 2) and has a parallel development of the functionalities and the safety analysis.

9.2 Directions for Future Work

Despite the advancements resulted from this research, we still see room for improvement. The improvements can be seen in a few directions: First, on the short term, we can consider that the use of patterns may not be only limited to architectural design but can be extended to (safety) analysis patterns as well. Second, also in short term, the methods disused in this thesis can be applied to other quality domains and standards. We can apply our method on existing standards such as ISO 21434 [56] for road vehicles cybersecurity as well as standards that are currently under development such as ISO/PAS 21448 [55] on the safety of the intended functionality (SOTIF). The topic of SOTIF is specially interesting given the focus of this standard on highly automated driving systems. Finally, on a longer term, the software tooling that is central for the effective application of the model-based approaches needs to be improved in terms of both maturity and integration with existing tooling.

[1] Here we use only a code name due to customer confidentiality issues

Using Patterns for Safety Analysis In Chapters 5.1 and 6 we showed the application of architecture patterns for designing safety-critical automotive applications. Design patterns are useful since they offer a solution for recurring challenges. The possibility of using patterns for the construction of safety cases is shown in previous research [60]. We can consider a similar approach towards safety analysis, such as hazard analysis and fault tree analysis.

Investigation of the patterns may result in more opportunities for the automation of safety analysis, provided an effective use of software tools as mentioned above. The value of these patterns can be paralleled to the value of templates; these patterns may be used for guiding users through the safety analysis, resulting in a more efficient way of working.

Application to Other Domains In Chapters 3 and 4, we applied conceptual modeling for functional safety and showed the feasibility of compliance checking with model-based approaches. One of the advantages of our method is that there are no dependencies on the safety domain; therefore, they can be used for other quality attributes such as cybersecurity. So long as there is a written source of information that industry has consensus upon, we can create a usable conceptual model for that industry.

The added value of these conceptual models becomes more visible once the integrated assurance of the various quality aspects, such as safety, security, and reliability, becomes critical due to increasing dependency of the automotive system to external ICT systems through numerous communication technologies.

Another foreseeable use of conceptual modeling is for the specification of the standards. The increasing number of norms and standards increases the risk of misalignment among various norms resulting in possible conflicting objectives for the organizations and companies following those norms. Conceptual modeling offers the means for reducing this risk.

Such an opportunity is presented currently with the development of ISO/PAS 21448 [55] on the safety of the intended functionality. This standard is closely related to the ISO 26262, it views functional safety with an outside-in perspective and intends to be complementary to the existing norms. This presents the challenge for recognizing possible overlaps and clarifying the differences. Conceptual modeling offers simple solutions for such challenges.

Software Tool We showed the feasibility of integrating model-based approach with existing software tools in Chapter 4. We show that by interpreting information from a typical document-based tool such as MS Excel, it is feasible to check for constraints and formulate feedback for the user (e.g., system developer). The scope of our research has limited the effectiveness of our proposed tool. Our proposed tooling may be extended in several manners.

First, we can extend the depths of the constraint checks. The current setup enables only simple constraints such as checks on existing relations. Further research is required for extending checks on more involved constraints such as inconsistencies between requirements. Such constraints require the use of natural language processing to formalize a statement regarding the content of system specification. Such statements need formalized project specific models that define the semantics of the design artifacts.

Second, the scope of the tool support may be extended horizontally by covering other safety analysis, such as Failure Mode and Effect Analysis (FMEA). If we extend the scope of the domain model to cover other parts of the ISO 26262 standard, we may use the same setup for analysis on the respective part.

Finally, we need better integration with other existing software tools for more impact on our approach. There are already many tools that support model-based development; therefore, we believe that the future software upgrade does not require to add another interface for the users. Rather, effective integration of model-based techniques with existing tools can be more valuable and bring the focus on information.

Bibliography

[1] A. Amroush, “Design Patterns for Safety-Critical Embedded Systems”, Ph.D. dissertation, Aachen University, 2010, p. 384, ISBN: 9781856177078. DOI: 10.1016/B978-1-85617-707-8.00006-6.

[2] E. Armengaud, Q. Bourrouilh, G. Griessnig, H. Martin, and P. Reichenpfader, “Using the CESAR Safety Framework for Functional Safety Management in the Context of ISO 26262”, Embedded Real Time Software and Systems, 2012.

[3] Audi, BMW, Daimler, Porsche, and VW, “Standardized E-Gas Monitoring Concept for Gasoline and Diesel Engine Control Units”, Tech. Rep., 2013.

[4] M. Batteux, T. Prosvirnova, A. Rauzy, and L. Kloul, “The AltaRica 3.0 project for model-based safety assessment”, in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, vol. 46, IEEE, Jul. 2013, pp. 741–746, ISBN: 978-1-4799-0752-6. DOI: 10.1109/INDIN.2013.6622976. [Online]. Available: http://ieeexplore.ieee.org/document/6622976/.

[5] S. Baumgart, J. Fröberg, and S. Punnekkat, “Analyzing hazards in system-of-systems: Described in a quarry site automation context”, in *11th Annual IEEE International Systems Conference, SysCon 2017 - Proceedings*, 2017, pp. 1–8, ISBN: 9781509046225. DOI: 10.1109/SYSCON.2017.7934783.

[6] K. Beckers, T. Frese, D. Hatebur, M. Heisel, I. Côté, T. Frese, D. Hatebur, and M. Heisel, “A structured and systematic model-based development method for automotive systems, considering the OEM/supplier interface”, *Reliability Engineering and System Safety*, vol. 158, no. September 2016, pp. 172–184, 2017, ISSN: 0951-8320. DOI: 10.1016/j.ress.2016.08.018. [Online]. Available: http://dx.doi.org/10.1016/j.ress.2016.08.018.

[7] G. Biggs, T. Sakamoto, and T. Kotoku, “A profile and tool for modelling safety information with design information in SysML”, *Software & Systems Modeling*, pp. 147–178, 2016, ISSN: 1619-1366. DOI: 10.1007/s10270-014-0400-x. [Online]. Available: http://dx.doi.org/10.1007/s10270-014-0400-x.

[8] T. Bijlsma and T. Hendriks, “A fail-operational truck platooning architecture”, *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 1819–1826, 2017. DOI: 10.1109/IVS.2017.7995970.

[9] T. Bijlsma, T. Hendriks, J. Vissers, L. Elshof, T. Jansen, and B. Krosse, “In-vehicle architectures for truck platooning: The challenges to reach sae automation level 3”, in *Proceedings of the 23rd ITS World Congress, 10–14 October 2016*, Melbourne, Australia, 2016.

[10] P. Bishop and R. Bloomfield, “A Methdology for Safety Case Development”, *Industrial Perspectives of Safety-critical Systems, P194-203*, 1998, Cited by 201.

[11] M. Born, J. Favaro, and O. Kath, “Application of ISO DIS 26262 in practice”, in *Proceedings of the 1st Workshop on Critical Automotive Applications: Robustness & Safety, CARS ’10*, Valencia, Spain: ACM, 2010, pp. 3–6, ISBN: 9781605589152. DOI: 10.1145/1772643.1772645.

[12] J. Brunel, P. Feiler, J. Hugues, B. Lewis, T. Prosvirnova, C. Seguin, and L. Wrage, “Performing Safety Analyses with AADL and AltaRica”, in *The 5th International Symposium on Model Based Safety Assessment (IMBSA 2017)*, vol. 2017, Trento, Italy, 2017, pp. 67–81, ISBN: 9783319641195. DOI: 10.1007/978-3-319-64119-5_5. [Online]. Available: http://link.springer.com/10.1007/978-3-319-64119-5%7B%5C_%7D5.

[13] F. Bushmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal, *Pattern-Oriented Software Architecture*. 1996, vol. 1, p. 476, ISBN: 0471958697. DOI: 10.1192/bjp.108.452.101.

[14] D. Cancila, F. Terrier, F. Belmonte, H. Dubois, H. Espinoza, S. Gérard, and A. Cuccuru, "SOPHIA: A modeling language for model-based safety engineering", *MODELS'09 ACES-MB WORKSHOP PROCEEDINGS*, vol. 507, pp. 11–25, 2009, ISSN: 16130073.

[15] J. S. Carroll, "Safety culture as an ongoing process : Culture surveys as opportunities for enquiry and change", *Work & Stress*, vol. 8373, pp. 37–41, 1998, ISSN: 0267-8373. DOI: 10.1080/02678379808256866.

[16] M. D. Cooper, "Towards a model of safety culture", *Safety Science*, vol. 36, no. 2, pp. 111–136, 2000, ISSN: 09257535. DOI: 10.1016/S0925-7535(00)00035-7.

[17] DAF Trucks N. V., *DAF Trucks*, 2015. [Online]. Available: http://www.daf.com/about-daf/daf-trucks-nv%7B%5C#%7D.

[18] S. Dariusz, D. Bert, D. Yoann, and V. V. Marc, "MODEL-BASED AND SCALABLE FUNCTIONAL SAFETY ENGINEERING METHODOLOGY FOR ON- AND OFF-HIGHWAY VEHICLES",

[19] E. de Gelder, A. Khabbaz Saberi, and H. Elrofai, "A method for scenario risk quantification for automated driving systems", in *The 26th International Technical Conference and exhibition on the Enhanced Safety of Vehicles (ESV)*, Eindhoven, The Netherlands, June 2019.

[20] G. Despotou, R. Alexander, and T. Kelly, "Addressing challenges of hazard analysis in systems of systems", *2009 3rd Annual IEEE Systems Conference*, pp. 167–172, 2009. DOI: 10.1109/SYSTEMS.2009.4815793.

[21] A. L. J. Dominguez, "Detection of Feature Interactions in Automotive Active Safety Features", p. 245, 2012.

[22] B. P. Douglass, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*. Addison-Wesley Professional, 2002, ISBN: 0201699567.

[23] *Eclipse Process Framework Project*, http://www.eclipse.org/epf/, 2015.

[24] ERTRAC Task Force: Connectivity and Automated Driving, "Automated Driving Roadmap", ERTRAC, Tech. Rep., Jan. 2015. DOI: 10.3141/2416-08. [Online]. Available: http://www.ertrac.org/uploads/documentsearch/id38/ERTRAC%7B%5C_%7DAutomated-Driving-2015.pdf.

[25] H. Espinoza, D. Cancila, B. Selic, and S. Gérard, "Challenges in combining SysML and MARTE for model-based design of embedded systems", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5562 LNCS, 2009, pp. 98–113, ISBN: 3642026737. DOI: 10.1007/978-3-642-02674-4_8.

[26] European Committee for Electrotechnical Standardization (CENELEC), *EN50126: Railway Applications–The Specification and Demonstration of Reliability Availability, Maintainability and Safety (RAMS)*, 1999.

[27] European Committee for Electrotechnical Standardization (CENELEC), *EN50128: Railway Applications-Communication, Signaling and Processing Systems-Software for Railway Control and Protection Systems*, 2011.

[28] European Committee for Electrotechnical Standardization (CENELEC), *EN50129: Railway Application–Safety Related Electronic Systems for Signaling*, 2000.

[29] M. Filax, T. Gonschorek, and F. Ortmeier, "Building Models we can rely on : Requirements Traceability for Model-based Verification Techniques", in *International Symposium on Model-Based and Assessment*, M. Bozzano and Y. Papadopoulos, Eds., Trento, Italy: Springer International Publishing, 2017, pp. 3–18. [Online]. Available: https://link.springer.com/chapter/10.1007%7B%5C%%7D2F978-3-319-64119-5%7B%5C_%7D1.

[30] M. Fleming, "Safety culture maturity model", The Keil Centre, Tech. Rep., 2001, 12 pages. DOI: ISBN0717619192. [Online]. Available: www.hse.gov.uk/research/.

[31] F. Franco, M. Mauro, S. Stevan Jr., and A. B. Lugli, "Model-Based Functional Safety for the Embedded Software of Automobile Power Window System", 2014.

[32] B. Freimut, T. Punter, S. Biffl, and M. Ciolkowski, *State-of-the-Art in Empirical Studies*. Virtuelles Software Engineering Kompetenzzentrum (ViSEK), 2002, pp. 1–108.

[33] E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, J. Ploeg, L. Friedman, and B. De Schutter, “Ontology for scenarios for the assessment of automated vehicles”, Submitted to Transportation Research Part C: Emerging Technologies. [Online]. Available: `https://arxiv.org/abs/2001.11507`.

[34] A. I. Glendon and N. A. Stanton, “Perspectives on safety culture”, *Safety Science*, vol. 34, no. 1-3, pp. 193–214, 2000, ISSN: 09257535. DOI: `10.1016/S0925-7535(00)00013-8`.

[35] Global Aviation Safety Network, *Operator’s flight safety handbook*. 2001, p. 180, ISBN: 9783486717631. DOI: `10.1524/9783486717631`.

[36] W. Goodall, T. Dovey, J. Bornstein, and B. Bonthron, “The rise of mobility as a service”, *Deloitte Review*, no. 20, pp. 111–130, 2017, ISSN: 00130613.

[37] R. Gordon and B. Kirwan, “Developing a safety culture in a research and development environment: Air Traffic Management domain.”, *Europe chapter of the Human Factor and Ergonomic Society conference*, vol. 129, p. 2865, 2004. [Online]. Available: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.9628%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf`.

[38] P. Graydon, “Towards a Clearer Understanding of Context and Its Role in Assurance Argument Confidence”, English, in *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, vol. 8666, Springer International Publishing, 2014, pp. 139–154.

[39] F. Guldenmund, “The nature of safety culture: a review of theory and research”, *Safety Science*, vol. 34, no. 1-3, pp. 215–257, 2000, ISSN: 09257535. DOI: `10.1016/S0925-7535(00)00014-X`.

[40] R. Hawkins, T. Kelly, J. Knight, and P. Graydon, “A New Approach to Creating Clear Safety Arguments”, in *Advances in systems safety*, Springer, 2011, pp. 3–23.

[41] S. Hecker and L. Goldenhar, “Understanding Safety Culture and Safety Climate in Construction : Existing Evidence and a Path Forward”, in *Safety Culture/Climate Workshop*, Washington, DC: The Center for Construction Research and Training. CPWR, 2014.

[42] M. Hillenbrand, M. Heinz, D. M. Klaus, N. Adler, J. Matheis, and C. Reichmann, “An Approach for Rapidly Adapting the Demands of ISO / DIS 26262 to Electric / Electronic Architecture Modeling”, in *Proceedings of 2010 21st IEEE International Symposium on Rapid System Protyping*, Fairfax, VA, 2010, pp. 1–7. DOI: `10.1109/RSP.2010.5656336`. [Online]. Available: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=%7B%5C&%7Darnumber=5656336%7B%5C&%7Disnumber=5656325`.

[43] *HSE: Safety Case Assessment Manual*, `http://www.hse.gov.uk/gas/supply/gasscham/gsmrscham.pdf`, Jun. 2011.

[44] P. Hudson, “Safety Culture - Theory and Practice”, in *The Human Factor in System Reliability - Is Human Performance Predictable*, 1999.

[45] P. Hudson, “Safety management and safety culture the long, hard and winding road”, *Occupational Health & Safety Management Systems Proceedings of the First National Conference*, p. 3, 2001. [Online]. Available: `http://www.ohs.com.au/ohsms-publication.pdf%7B%5C#%7Dpage=11`.

[46] P. Hudson, “Safety culture: the ultimate goal”, *Flight Safety Australia*, no. October, pp. 29–31, 2001. [Online]. Available: `http://82.94.179.196/bookshelf/books/1091.pdf`.

[47] IEC, *IEC 61508-1: Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements*. 2010.

[48] D. M. of Infrastructure, *European Truck Platooning Challeng*, 2016. [Online]. Available: `https://www.eutruckplatooning.com/default.aspx` (visited on 08/24/2016).

[49] Institute Ford Design, *Ford Failure Mode and Effects Analysis*. 2004, p. 290.

[50] International Organization of Motor Vehicle Manufacturers (OICA), *OICA*, 2019. [Online]. Available: `http://www.oica.net/category/safety/global-safety/`.

[51] ISO, “IEC 25010: 2011”, *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models*, 2011.

[52] ISO, *ISO 26262: Road vehicles - Functional safety*. Geneva, Switzerland: International Organization for Standardization, 2011.

[53] ISO, *ISO/AWI PAS 21448: Road vehicles – Safety of the intended functionality*. Geneva, Switzerland: International Organization for Standardization, 2017.

[54] ISO, *ISO/DIS 26262: Road Vehicles – Functional safety*. Geneva, Switzerland: International Organization for Standardization, 2018.

[55] ISO, "ISO/PAS 21448: Road vehicles — Safety of the intended functionality", International Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2019.

[56] ISO, *ISO/SAE CD 21434*. Geneva, Switzerland: International Organization for Standardization, 2019.

[57] "ISO/IEC/IEEE 42010: Systems and software engineering - Architecture description", *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pp. 1–46, 2011. DOI: 10.1109/IEEESTD.2011.6129467.

[58] T. Kelly and J. McDermid, "Safety Case Patterns - Reusing Successful Arguments", in *IEEE Colloquium on Understanding Patterns and Their Application to Systems Engineering (Digest No. 1998/308)*, cited by 41, Apr. 1998, pp. 3/1–3/9. DOI: 10.1049/ic:19980543.

[59] T. Kelly and R. Weaver, "The Goal Structuring Notation - A Safety Argument Notation", *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases*, 2004, Cited by 257.

[60] T. P. Kelly and J. A. McDermid, "Safety case construction and reuse using patterns", in *Safe Comp 97*, P. Daniel, Ed., London: Springer London, 1997, pp. 55–69, ISBN: 978-1-4471-0997-6.

[61] A. Khabbaz Saberi, A. Smulders, and J. J. Lukkien, "Towards a holistic assurance methodology: From component to information assurance", in *The Fast Abstract Track of the 38th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2019)*, Finland, 2019.

[62] A. Khabbaz Saberi, F. Benders, R. Koch, J. Lukkien, and M. van den Brand, "A method for quantitative measurement of safety culture based on iso 26262", English, in *Evolution of System Safety*, M. Parsons and T. Kelly, Eds., CreateSpace Independent Publishing Platform, Feb. 2018, pp. 203–218, ISBN: 1979733619.

[63] A. Khabbaz Saberi, Y. Luo, F. Pawel Cichosz, M. van den Brand, and S. Jansen, "An approach for functional safety improvement of an existing automotive system", in *9th Annual IEEE International Systems Conference, SysCon 2015 - Proceedings*, 2015, ISBN: 9781479959273. DOI: 10.1109/SYSCON.2015.7116764.

[64] A. Khabbaz Saberi, E. Barbier, F. Benders, and M. Van Den Brand, "On functional safety methods: A system of systems approach", English, in *12th Annual IEEE International Systems Conference, SysCon 2018 - Proceedings*, United States: Institute of Electrical and Electronics Engineers (IEEE), May 2018. DOI: 10.1109/SYSCON.2018.8369598.

[65] A. Khabbaz Saberi, J. Hegge, T. Fruehling, and J. F. Groote, "Beyond SOTIF: Black Swans and Formal Methods", in *14th Annual IEEE International Systems Conference (SysCon 2020)*.

[66] A. Khabbaz Saberi, Y. Luo, F. P. Cichosz, M. van den Brand, S. Jansen, F. Pawel Cichosz, M. van den Brand, and S. Jansen, "An Approach for Functional Safety Improvement of an Existing Automotive System", in *9th Annual IEEE International Systems Conference (SysCon 2015)*, 2015, pp. 277–282, ISBN: 9781479959273. DOI: 10.1109/SYSCON.2015.7116764.

[67] A. Khabbaz Saberi, D. van den Brand, and M. van den Brand, "Towards compliance assurance for automotive safety-critical development: a model-based approach", in *The Poster Session of 6th International Symposium on Model-Based Safety and Assessment (IMBSA 2019)*, Thessaloniki, Greece, 2019.

[68] A. Khabbaz Saberi, J. Vissers, and F. Benders, "On the Impact of Early Design Decisions on Quality Attributes of Automated Driving Systems", in *13th Annual IEEE International Systems Conference (SysCon 2019)*, Orlando, Florida, USA.

[69] A. Knoll, C. Buckl, K.-J. Kuhn, and G. Spiegelberg, "The RACE Project: An Informatics-Driven Greenfield Approach to Future E/E Architectures for Cars", in *Automotive Systems and Software Engineering*, Y. Dajsuren and M. van den Brand, Eds., Springer International Publishing, 2019, ISBN: 978-3-030-12156-3. DOI: 10.1007/978-3-030-12157-0.

[70] M. Krammer, E. Armengaud, and Q. Bourrouilh, "Method Library Framework for Safety Standard Compliant Process Tailoring", in *37th EUROMICRO Conference on Software Engineering and Advanced Applications*, IEEE, 2011, pp. 302–305.

[71] G. Kristen, *Object Orientation - The KISS Method, From Information Architecture to Information System*. Massachusetts, USA: Addison-Wesley, 1994, ISBN: 0201422999.

[72] J. Langheim, B. Guegan, L. Maillet-Contoz, K. Maaziz, G. Zeppa, F. Phillipot, S. Boutin, I. Aboutaleb, and P. David, "System Architecture, Tools and Modelling for Safety Critical Automotive Applications - The R&D Project SASHA", in *ERTS2 2010, Embedded Real Time Software & Systems*, Toulouse, France, 2010, pp. 1–8.

[73] A. Legendre, A. Lanusse, A. Rauzy, A. Legendre, A. Lanusse, A. Rauzy, T. Model, and S. Between, "Toward Model Synchronization Between Safety Analysis and System Architecture Design in Industrial Contexts", in *International Symposium on Model-Based and Assessment*, M. Bozzano and Y. Papadopoulos, Eds., Trento, Italy: Springer International Publishing, 2017, pp. 35–49, ISBN: 9783319641188.

[74] Z. Li, "A Systematic Approach and Tool Support for Assessing GSN-based Safety Case", M.S. thesis, Eindhoven University of Technology, 2016, p. 97.

[75] J. J. Lukkien, "A Systems of Systems perspective on the Internet of Things", *Special Issue on 8th International Workshop on Compositional Theory and Technology for Real-Time Embedded Systems (CRTS 2015)*, vol. 13, 2016.

[76] Y. Luo, "From Conceptual Models to Safety Assurance", Ph.D. dissertation, Eindhoven University of Technology, 2016, ISBN: 978-3-319-12206-9. DOI: 10.1007/978-3-319-12206-9_16.

[77] Y. Luo, M. van den Brand, L. Engelen, and M. Klabbers, "A modeling approach to support safety assurance in the automotive domain", English, in *Progress in Systems Engineering*, ser. Advances in Intelligent Systems and Computing, vol. 1089, Springer International Publishing, 2015, pp. 339–345.

[78] Y. Luo, M. van den Brand, L. Engelen, and M. Klabbers, "From Conceptual Models to Safety Assurance", English, in *Conceptual Modeling*, ser. Lecture Notes in Computer Science, vol. 8824, Springer International Publishing, 2014, pp. 195–208.

[79] Y. Luo, A. Khabbaz Saberi, T. Bijlsma, J. J. Lukkien, and M. van den Brand, "An architecture pattern for safety critical automated driving applications: Design and analysis", in *2017 Annual IEEE International Systems Conference (SysCon)*, 2017, pp. 1–7, ISBN: 9781509046225. DOI: 10.1109/SYSCON.2017.7934739.

[80] Y. Luo, A. Khabbaz Saberi, and M. van den Brand, "Safety-Driven Development and ISO 26262", in *Automotive Systems and Software Engineering*, Y. Dajsuren and M. van den Brand, Eds., Springer International Publishing, 2019, ISBN: 978-3-030-12156-3. DOI: 10.1007/978-3-030-12157-0.

[81] Y. Luo, M. G. J. van den Brand, L. Engelen, and M. Klabbers, "A Modeling Approach to Support Safety Assurance in the Automotive Domain", in *Progress in Systems Engineering*, vol. 1089, Springer International Publishing, 2015, pp. 339–345.

[82] Y. Luo, M. G. J. van den Brand, and A. Kiburse, "Safety Case Development with SBVR-based Controlled Language", in *Proceedings of Third International Conference on Model-Driven Engineering and Software Development*, 2015.

[83] Y. Luo, M. Van den Brand, L. Engelen, J. M. Favaro, M. Klabbers, and G. Sartori, "Extracting Models from ISO 26262 for Reusable Safety Assurance", in *Safe and Secure Software Reuse - 13th International Conference on Software Reuse*, vol. 7925, Springer Berlin Heidelberg, 2013, pp. 192–207.

[84] Y. Luo, M. van den Brand, Z. Li, and A. Khabbaz Saberi, "A systematic approach and tool support for GSN-based safety case assessment", *Journal of Systems Architecture*, vol. 76, pp. 1–16, 2017, ISSN: 13837621. DOI: 10.1016/j.sysarc.2017.04.001.

[85] P. Mauborgne, S. Deniaud, E. Levrat, E. Bonjour, J. Micaelli, and D. Loise, "Operational and System Hazard Analysis in a Safe Systems Requirement Engineering Process - Application to automotive industry", *Safety Science*, vol. 87, pp. 256–268, 2016, ISSN: 18791042. DOI: 10.1016/j.ssci.2016.04.011.

[86] P. Mauborgne, S. Deniaud, É. Levrat, É. Bonjour, J.-p. Micaëlli, and D. Loise, "The Determination of Functional Safety Concept coupled with the definition of coupled with the definition of Logical Architecture : a framework of analysis from the automotive industry", in *20th IFAC World Congress, IFAC 2017*, Toulouse, France, 2017, pp. 7549–7554.

[87] G. Me, C. Calero, and P. Lago, "Architectural patterns and quality attributes interaction", in *Qualitative Reasoning about Software Architectures (QRASA)*, Venice, Italy, 2016, pp. 27–36.

[88] G. Meyer, J. Dokic, B. Müller, and G. Meyer, "European Roadmap Smart Systems for Automated Driving", Tech. Rep., 2015.

[89] G. Meyer and S. Shaheen, *Disrupting Mobility*, G. Meyer and S. Shaheen, Eds., ser. Lecture Notes in Mobility. Springer International Publishing, 2017, ISBN: 978-3-319-51601-1. DOI: 10.1007/978-3-319-51602-8. [Online]. Available: http://link.springer.com/10.1007/978-3-319-51602-8.

[90] Ministry of Infrastructure and the Environment, *Mobility, public transport and road safety*, 2015. [Online]. Available: http://www.government.nl/issues/mobility-public-transport-and-road-safety/road-safety.

[91] *Defence Standard 00-55 Part 1*, http://www.software-supportability.org/Docs/00-55_Part_1.pdf, 1997.

[92] NHTSA, "TRAFFIC SAFETY FACTS", no. February, pp. 2–3, 2015.

[93] E. van Nunen, F. Esposto, A. Khabbaz Saberi, and J. P. Paardekooper, "Evaluation of safety indicators for truck platooning", in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2017, pp. 1013–1018, ISBN: 9781509048045. DOI: 10.1109/IVS.2017.7995847.

[94] OMG, *SBVR: Semantics Of Business Vocabulary And Rules (version 1.2)*, Sep. 2013.

[95] OMG, *Software and Systems Process Engineering Metamodel Specification*, http://www.omg.org/spec/SPEM/2.0/, Apr. 2008.

[96] *OPENCOSS: Deliverable D2.2 - High-level requirements (report)*, http://www.opencoss-project.eu/node/7, 2013.

[97] B. Orlic, R. Mak, I. David, and J. Lukkien, "Concepts and diagram elements for architectural knowledge management", *Proceedings of the 5th European Conference on Software Architecture - ECSA '11*, p. 1, 2011. DOI: 10.1145/2031759.2031763. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2031759.2031763.

[98] O. Örsmark, *Will your safety case pass an ISO 26262 assessment?*, http://safety.addalot.se/2015/programme, Mar. 2015.

[99] R. Panesar-Walawege, M. Sabetzadeh, and L. Briand, "Using UML Profiles for Sector-Specific Tailoring of Safety Evidence Information", in *30th ACM International Conference on Conceptual Modeling (ER)*, ser. Lecture Notes in Computer Science, M. Jeusfeld, L. Delcambre, and T.-W. Ling, Eds., vol. 6998, Heidelberg: Springer, 2011, pp. 362–378.

[100] K. Philippe, L. Patricia, and v. V. Hans, "Building up and reasoning about architectgural knowledge", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4214, no. DECEMBER, pp. 95–110, 2006, ISSN: 03029743. DOI: 10.1007/11921998. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-70350005558%7B%5C&%7DpartnerID=tZOtx3y1.

[101] M. Piers, C. Montijn, and A. Balk, "Safety Culture Framework for the ECAST SMS-WG", no. March, pp. 1–14, 2009.

[102] J. Ploeg, *Analysis and design of controllers for cooperative and automated driving*. 2014, p. 183, ISBN: 9789462591042. DOI: 10.6100/IR772224.

[103] D. Potop Butucaru, *ASSUME*, 2018. [Online]. Available: https://itea3.org/project/assume.html.

[104] T. Prosvirnova, E. Saez, C. Seguin, and P. Virelizier, "Handling consistency between safety and system models", in *International Symposium on Model-Based and Assessment*, M. Bozzano and Y. Papadopoulos, Eds., Trento, Italy: Springer International Publishing, 2017, pp. 19–34. [Online]. Available: https://link.springer.com/chapter/10.1007%7B%5C%%7D2F978-3-319-64119-5%7B%5C_%7D2.

[105] S. Pugh, *Total design: integrated methods for successful product engineering*. Addison-Wesley Wokingham, 1991.

[106] RDW, *About RDW*, 2015. [Online]. Available: https://www.rdw.nl/overrdw/Paginas/default.aspx.

[107] Ricardo GmbH, *Ricardo GmbH*, 2015. [Online]. Available: http://www.ricardo.com/.

[108] *ROADART*. [Online]. Available: http://www.roadart.eu/about.

[109] K. Rumar, “The Role of Perceptual and Cognitive Filters in Observed Behavior”, in *Human Behavior and Traffic Safety*, L. Evans and R. Schwing, Eds., Springer US, 1985, pp. 151–170, ISBN: 978-1-4612-9280-7, 978-1-4613-2173-6. DOI: 10.1007/978-1-4613-2173-6_8. [Online]. Available: http://link.springer.com/chapter/10.1007/978-1-4613-2173-6%7B%5C_%7D8%7B%5C%%7D5Cnhttp://link.springer.com/chapter/10.1007/978-1-4613-2173-6%7B%5C_%7D8.

[110] B. Rumpe, *Modeling with UML: Language, Concepts, Methods*, 1st. Springer Publishing Company, Incorporated, 2016, ISBN: 331933932X, 9783319339320.

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

[112] SAE International, *SAE Six Levels of Automation*, 2014. [Online]. Available: http://www.sae.org/misc/pdfs/automated%7B%5C_%7Ddriving.pdf.

[113] SAE International, “Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems”, *SAE International*, vol. J3016, pp. 1–12, 2014.

[114] *Safety Case Repository*, http://dependability.cs.virginia.edu/info/Safety_Cases:Repository, 2013.

[115] A. Salikiryaki, I. Petrova, and S. Baumgart, “Graphical Approach for Modeling of Safety and Variability in Product Lines”, in *41st Euromicro Conference on Software Engineering and Advanced Applications*, 2015. [Online]. Available: http://www.es.mdh.se/publications/4038-.

[116] M. R. Sena Marques, E. Siegert, and L. Brisolara, “Integrating UML, MARTE and sysml to improve requirements specification and traceability in the embedded domain”, in *Proceedings - 2014 12th IEEE International Conference on Industrial Informatics, INDIN 2014*, 2014, pp. 176–181, ISBN: 9781479949052. DOI: 10.1109/INDIN.2014.6945504.

[117] A. A. Shah, A. A. Kerzhner, D. Schaefer, and C. J. Paredis, “Multi-view modeling to support embedded systems engineering in SysML”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5765 LNCS, 2010, pp. 580–601, ISBN: 3642173217. DOI: 10.1007/978-3-642-17322-6_25.

[118] M. Siegel, “The sense-think-act paradigm revisited”, *ROSE 2003 - 1st IEEE International Workshop on Robotic Sensing 2003: Sensing and Perception in 21st Century Robotics*, no. June, pp. 5–6, 2003. DOI: 10.1109/ROSE.2003.1218700.

[119] *Software Consideration in Airborne Systems and Equipment Certification: RTCA DO-178C*, Dec. 2011.

[120] SRR, “2016 Automotive Warranty and Recall Report: New Insights For the Road Ahead”, 2016.

[121] P. Sternudd, “Unambiguous Requirements in Functional Safety and ISO 26262: Dream or Reality?”, M.S. thesis, Uppsala University, 2011.

[122] D. Szymanski, M. Scharrer, and G. Macher, “Model-Based Functional Safety Engineering”, in *Comprehensive Energy Management - Safe Adaptation, Predictive Control and Thermal Management*, D. Watzenig and B. Brandstätter, Eds., SpringerBriefs in Applied Sciences and Technology, 2018, ISBN: 978-3-319-57444-8. DOI: 10.1007/978-3-319-57445-5. [Online]. Available: http://link.springer.com/10.1007/978-3-319-57445-5.

[123] B. Thalheim, “The Theory of Conceptual Models, the Theory of Conceptual Modelling and Foundations of Conceptual Modelling”, in *Handbook of Conceptual Modeling*, 2011, pp. 543–577, ISBN: 978-3-642-15864-3. DOI: 10.1007/978-3-642-15865-0. [Online]. Available: http://link.springer.com/10.1007/978-3-642-15865-0.

[124] K. Thramboulidis and S. Scholz, “Integrating the 3 + 1 SysML View Model with Safety Engineering”, in *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, 2010, pp. 13–16, ISBN: 9781424468508. DOI: 10.1109/ETFA.2010.5641353.

[125] TNO, *Automated Driving Two Trucks Enters a New Phase*, 2015. [Online]. Available: https://www.tno.nl/en/about-tno/news/2015/3/automated-driving-with-two-trucks-enters-a-new-phase/.

[126] TNO, *TNO*, 2016. [Online]. Available: https://www.tno.nl/en/about-tno/ (visited on 06/20/2004).

[127] M. van den Brand and J. F. Groote, “Software engineering: Redundancy is key”, *Science of Computer Programming*, vol. 97, pp. 75–81, 2013, ISSN: 01676423. DOI: 10.1016/j.scico.2013.11.020.

[128] E. van Nunen, F. Esposto, A. Khabbaz Saberi, and J. Paardekooper, "Evaluation of safety indicators for truck platooning", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 1013–1018. DOI: 10.1109/IVS.2017.7995847.

[129] J. L. de la Vara and R. K. Panesar-Walawege, "SafetyMet: A Metamodel for Safety Standards", in *MODELS 2013, LNCS 8107*, A. Moreira, Ed., Springer, Berlin, Heidelberg, 2013, pp. 69–86. DOI: 10.1007/978-3-642-41533-3_5. [Online]. Available: http://link.springer.com/10.1007/978-3-642-41533-3_5.

[130] R. Viereckl, A. Koster, E. Hirsh, and D. Ahlemann, "Opportunities , risk , and turmoil on the road to autonomous vehicles", PWC, Tech. Rep., 2016. [Online]. Available: https://www.strategyand.pwc.com/reports/connected-car-2016-study.

[131] K. Warszawska and A. Kraslawski, "Method for quantitative assessment of safety culture", *Journal of Loss Prevention in the Process Industries*, pp. 323–330, 2015, ISSN: 0950-4230. DOI: http://dx.doi.org/10.1016/j.jlp.2015.09.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950423015300309.

[132] R. Weissnegger, "Design and Verification Process for Safety-Critical Embedded Systems in the Automotive Domain", Ph.D. dissertation, Graz University of Technology, 2017.

[133] R. Weissnegger, C. Kreiner, and C. Steger, "A novel design method for automotive safety- critical systems based on uml / marte A Novel Design Method for Automotive Safety-Critical Systems based on UML / MARTE", in *2015 Forum on specification & Design Languages*, Barcelona, Spain, 2015.

[134] S. A. White, "Introduction to bpmn", *IBM Cooperation*, vol. 2, no. 0, p. 0, 2004.

[135] J. Wu, T. Yue, S. Ali, and H. Zhang, "A modeling methodology to facilitate safety-oriented architecture design of industrial avionics software", *Software: Practice and Experience*, vol. 45, no. 7, pp. 893–924, Jul. 2015, ISSN: 00380644. DOI: 10.1002/spe.2281. [Online]. Available: http://doi.wiley.com/10.1002/spe.2281.

[136] J.-B. Yang and M. G. Singh, "An Evidential Reasoning Approach for Multiple-attribute Decision Making with Uncertainty", *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 1, pp. 1–18, 1994.

[137] J.-B. Yang and D.-L. Xu, "On the Evidential Reasoning Algorithm for Multiple Attribute Decision Analysis Under Uncertainty", *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 32, no. 3, pp. 289–304, 2002.

[138] T. Yuan and T. Kelly, "Argument-based approach to computer system safety engineering", *International Journal of Critical Computer-Based Systems*, vol. 3, no. 3, pp. 151–167, 2012.

[139] T. Yuan, T. Kelly, T. Xu, H. Wang, and L. Zhao, "A dialogue-based safety argument review tool", 2013.

[140] G. Zoughbi, L. Briand, and Y. Labiche, "Modeling Safety and Airworthiness (RTCA DO-178B) Information: Conceptual Model and UML Profile", *Softw. Syst. Model.*, vol. 10, no. 3, pp. 337–367, Jul. 2011, ISSN: 1619-1366. DOI: 10.1007/s10270-010-0164-x.

Curriculum Vitae

Arash was born on 27-04-1988 in Tehran, Iran. After finishing B.Sc. in 2010 at Shahid Beheshti University in Tehran, Iran, he studied Master Program of Embedded Systems at Technical University of Eindhoven (TU/e) in The Netherlands. In 2013 he graduated within the Systems Control group on Control Relevant MIMO Parametric Identification. In 2015, he completed the Professional Doctorate in Engineering (PDEng) in Automotive System Design at TU/e. From 2015 he started a PhD project at TU/e in collaboration with the Integrated Vehicle Safety (IVS) Department of TNO, of which the results are presented in this dissertation. Since 2015 he is employed at TNO, IVS.

List of Publications

1. E. de Gelder, J. P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, J. Ploeg, L. Friedman, and B. De Schutter "Ontology for scenarios for the assessment of automated vehicles," (Submitted to Transportation Research Part C: Emerging Technologies)

2. A. Khabbaz Saberi, J. Hegge, T. Fruehling, and J.F. Groote "Beyond SOTIF: Black Swans and Formal Methods," *14th Annual IEEE International Systems Conference (SysCon 2020)* (Accepted)

3. A. Khabbaz Saberi, D. van den Brand, and M. G. J. van den Brand "Towards compliance assurance for automotive safety-critical development: a model-based approach," *in the poster session of the 6th International Symposium on Model-Based Safety and Assessment (IMBSA 2019),* 2019

4. A. Khabbaz Saberi, A. Smulders, and J. J Lukkien "Towards a Holistic Assurance Methodology: From Component to Information Assurance," *The Fast Abstract Track of the 38th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2019)*, Finland, 2019

5. E. de Gelder, A. Khabbaz Saberi, and H. Elrofai "A method for scenario risk quantification for automated driving systems," *The 26th International Technical Conference and exhibition on the Enhanced Safety of Vehicles (ESV)*, Eindhoven, The Netherlands, June 2019

6. A. Khabbaz Saberi, J. Vissers, and F. P. A. Benders "On the Impact of Early Design Decisions on Quality Attributes of Automated Driving Systems," *13th Annual IEEE International Systems Conference (SysCon 2019)*, April 2019

7. Y. Luo, A. Khabbaz Saberi, and M. G. J. van den Brand "Safety-Driven Development and ISO 26262," *in Automotive Systems and Software Engineering,* Springer International Publishing, 2019

8. A. Khabbaz Saberi, E. Barbier, F. Benders, and M. G. J. van den Brand "On functional safety methods: A system of systems approach," *12th Annual IEEE International Systems Conference (SysCon 2018)* April 2018

9. A. Khabbaz Saberi, F. Benders, R. Koch, J. J. Lukkien, and M. G. J. van den Brand "A method for quantitative measurement of safety culture based on ISO 26262," *Evolution of System Safety: Proceedings of the Twenty-Sixth Safety-Critical Systems Symposium, 6-8 February 2018*, York, United Kingdom, p. 203-218, 2018

10. Y. Luo, M. G. J. van den Brand, and A. Khabbaz Saberi "A systematic approach and tool support for GSN-based safety case assessment," *Journal of Systems Architecture: Embedded Software Design : the EUROMICRO*, p. 1-16, May 2017

11. E. van Nunen, F. Esposto, A. Khabbaz Saberi, and J. P. Paardekooper "Evaluation of safety indicators for truck platooning," *2017 IEEE Intelligent Vehicles Symposium (IV)*,Los Angeles, California, p. 1013-1018, June 2017

12. Y. Luo, A. Khabbaz Saberi, T. Bijlsma, J. J. Lukkien, and M. G. J. van den Brand "An architecture pattern for safety critical automated driving applications: design and analysis," *11th Annual IEEE International Systems Conference (SysCon 2017)*, 24-27 April 2017

13. A. Khabbaz Saberi, Y. Luo, F. P. Cichosz, M. G. J. van den Brand, and S. Jansen "An Approach for Functional Safety Improvement of an Existing Automotive System," *9th Annual IEEE International Systems Conference (SysCon 2015)*, 2015