Multi-Concerns Engineering for Safety-Critical Software Systems

Multi-Criteria Decision Making, Change Management and Variability

Dissertation

for the degree of

Doctor of Natural Sciences (Dr. rer. nat.)

Philipp Wilhelm Lohmüller

University of Augsburg

Department of Computer Science

Software Methodologies for Distributed Systems

November 2019

Multi-Concerns Engineering for Safety-Critical Software Systems

Supervisor:	**Prof. Dr. Bernhard Bauer**, Department of Computer Science University of Augsburg, Germany
Advisor:	**Prof. Dr. Robert Lorenz**, Department of Computer Science University of Augsburg, Germany
Thesis Defence:	Thursday, 20th February 2020

Abstract

Today, safety-critical systems can be found in various domains, including the automotive industry, avionics and the medical environment. The individual systems are cross-linked via wired or wireless interfaces to provide certain (comfort) functionalities. However, these functionalities often have weak points and thus allow access for third parties. Non circumventing these vulnerabilities can have fatal consequences for human lives, i.e. safety properties are violated. Therefore, it should be regarded as highest priority to ensure maximum safety. To achieve this, other concerns have to be considered, such as security (crime prevention) or timing (real-time requirements). In this context, we speak about Multi-Concerns (MC). However, the consideration of MC often leads to conflicts, as the following example shows: In the event of an accident, an airbag has to be able to be triggered immediately to protect the occupants from internal and external injuries (safety). In order to not manipulate the functionality of the airbag by third parties, a secure encryption of the data is necessary (security). In the event of a collision, however, it can result in the airbag not being triggered in time (timing), as decrypting the corresponding commands take too long due to a chosen encryption algorithm which has a security level too high. Thus, in this case the safety, the primary goal, would be in danger. To solve such conflicts, appropriate trade-offs have to be calculated which is the first objective of this thesis. Therefore, in this thesis an approach is presented to model MC and to calculate trade-offs. In this context, a risk assessment and mitigation technique are considered. A methodology is presented to determine from a set of possible alternative solutions the one that best meets the safety-critical requirements.

(Software) products change constantly over time and the development progresses steadily. In this context, the requirements of safety-critical components often change as well. This usually means that existing components of a system have to be replaced or added to meet the requirements. The calculation of optimal trade-offs of the system under investigation is impaired due to changed security requirements. It is therefore another objective of this thesis to identify the components affected by a change. It is important to ensure traceability throughout the entire process chain and to define well-defined impact rules.

To minimise the analysis of the overall system (software) products can be configured individually according to the needs of the customer to guarantee variability. In this context we often speak about Software Product Lines (SPLs) or variability. For instance, it is possible to customise a mid-range car from several million configuration options. Not only components such as paintwork, seats, etc. play a role, but also safety-critical components such as Advanced Driver Assistance Systems (ADASs) are taken into account. Each combination consisting of safety-critical components has different safety requirements which have to be taken into account. Due to the large number of possible combinations of safety-critical soft-

ware product lines it is not possible to check each variation individually. Therefore, it is a further goal of this thesis to model software product lines in a standardised way taking safety-critical requirements into account. The focus is on clustering software product line variations with identical safety and security requirements to efficiently calculate trade-offs and the impact of changes.

Zusammenfassung

Sicherheitskritische Systeme kommen heutzutage in unterschiedlichen Domänen vor, darunter in der Automobilbranche, Avionik-Bereich oder auch im medizinischen Umfeld. Dabei sind die einzelnen Systeme über kabelgebundene oder kabellose Schnittstellen vernetzt, um gewisse (Komfort-)Funktionalitäten zur Verfügung zu stellen. Jedoch weisen diese Funktionalitäten oft Schwachstellen auf und ermöglichen somit den Zugriff für Dritte. Das Vernachlässigen dieser Schwachstellen kann fatale Konsequenzen für Menschenleben haben, d.h. die Safety-Eigenschaften werden verletzt. Daher hat es höchste Priorität, maximale Safety zu gewährleisten. Um dies zu erreichen, müssen auch andere Concerns betrachtet werden, wie z.B. Security (Betriebssicherheit) oder Timing (z.B. Echtzeitanforderungen). Daher sprechen wir in diesem Zusammenhang auch von Multi-Concerns (MC) Die Betrachtung von MC führt jedoch oft zu Konflikten, die durch folgendes Beispiel verdeutlicht werden: Ein Airbag muss im Fall eines Unfalls sofort auslösen können, um die Insassen vor Verletzungen zu schützen (Safety). Um die Funktionalität des Airbags vor äußeren Angriffen zu schützen ist eine sichere Verschlüsselung der Daten vonnöten (Security). Dies kann jedoch im Falle eines Aufpralls dazu führen, dass der Airbag nicht rechtzeitig auslöst (Timing), da das Entschlüsseln der entsprechenden Befehle aufgrund eines zu sicher gewählten Verschlüsselungsalgorithmus zu lange dauert . Somit wäre in diesem Fall die Safety, das primäre Ziel, in Gefahr. Ziel dieser Arbeit ist es daher, entsprechende Trade-Offs zu berechnen, um solche Konflikte zu lösen. In dieser Dissertation wird ein Ansatz vorgestellt, um MC zu modellieren und Trade-Offs zu berechnen. In diesem Zusammenhang wird ein Verfahren zur Risikobewertung und Risikominimierung vorgestellt. Ebenso wird eine Methodik präsentiert, um aus einer Menge möglicher Alternativen diejenige zu bestimmen, welche den sicherheitskritischen Anforderungen am meisten gerecht wird.

(Software-)Produkte entwickeln sich stetig weiter und neue Features werden realisiert. In diesem Zusammenhang ändern sich auch oft die Anforderungen sicherheitskritischer Komponenten. Dies hat meist zur Folge, dass bestehende Komponenten eines Systems ausgetauscht oder ergänzt werden müssen, um den gestellten Anforderungen gerecht zu werden. Die Berechnung von optimalen Trade-Offs des zu untersuchenden Systems wird aufgrund von geänderten Sicherheitsanforderungen beeinträchtigt. Es ist daher ein weiterer Fokus dieser Arbeit, die Komponenten zu identifizieren, die von einer Änderung betroffen sind. Dabei ist es wichtig, Traceability über den kompletten Verlauf der Prozesskette zu gewährleisten und wohl definierte Auswirkungsregeln zu definieren.

Heutzutage können (Software-)Produkte individuell nach den Bedürfnissen des Kunden konfiguriert werden, um Variabilität zu gewährleisten. In diesem Zusammenhang sprechen wir von (Software-)Produktlininen. Es ist z.B. möglich, sich einen Mittelklassewagen aus mehreren Millionen Konfigurationsmöglichkeiten

zu individualisieren. Dabei spielen nicht nur Komponenten wie Lackierung, Sitze, etc. eine Rolle, sondern es werden auch sicherheitskritische Komponenten wie Fahrerassistenzsysteme berücksichtigt. Jede Kombination mit sicherheitskritischen Komponenten hat dabei unterschiedliche Sicherheitsanforderungen, die berücksichtigt werden müssen. Aufgrund der Vielzahl an Kombinationsmöglichkeiten von sicherheitskritischen Software-Produktlinien ist es nicht möglich jede Variation einzeln zu überprüfen. Ziel dieser Arbeit ist daher auch Software-Produktlinien unter Berücksichtigung sicherheitskritischer Anforderungen standardisiert zu modellieren. Der Fokus liegt darin, Variationen von Software-Produktlinien mit identischen Sicherheitsanforderungen zu gruppieren, um anschließend effizient Trade-Offs und die Auswirkungen von Änderungen berechnen zu können.

Acknowledgements

This thesis has been written during my employment as research assistant and doctoral student at the SMDS lab of the University of Augsburg. I would like to use this space to thank all the people who have contributed to the completion of this thesis.

First of all I would like to thank my supervisor Prof. Dr. Bernhard Bauer who provided me the opportunity to write this PhD thesis at his professorship. Thank you, Bernhard, that you always took the time for numerous discussions and the esteemed feedback to finalise this dissertation. In the first years, when I tried to concretise the subject of this thesis I was sometimes discouraged. You and the family working atmosphere motivated me constantly to pursue the topic. Further thanks go to Prof. Dr. Robert Lorenz who accepted to be advisor of this thesis. I know it takes a lot of time to read and peer review such a thesis.

I would like to thank all my present and former colleagues of the SMDS lab. I really appreciated the harmonic working environment and enjoyed the discussions during the lunch breaks. At this point I would like to name and thank Andrea Fendt and Julia Rauscher for numerous technical discussions and the opportunity to write conference papers together. Many thanks also to my present and former office colleagues Christoph Etzel, Reinhard Pröll and Adrian Rumpold for technical as well as pleasant conversations between the screens. Sonja, I am very grateful to you that you completed many organisational matters of mine in the last years and thus to exonerate me therefrom.

I would like to thank my family and all my friends. You have always managed to clear my mind even in stressful days. I really enjoyed all the common mountain hikes, sports activities, day trips or gaming nights. Thank you, Simon Lohmüller and Emanuel Wortberg, for taking your precious time to proofread this PhD thesis. A hearty thank goes to my parents Martina and Gerald who always supported me during my education and study. Without their dedication and motivation in all decisions this thesis would not exist. I am also grateful to my brother Simon who has always advised me well and cheered me up during the last years. Last but not least I would like to thank my girlfriend Melanie who always believed in me when I had doubts about this thesis and encouraged me all the time. Thank you for standing by my side.

Contents

Part I.

Introduction and Basics

1 Introduction

The aim of this chapter is to provide a motivation and problem outline regarding Safety-Critical Systems (SCSs). In particular, it is motivated by means of examples of the automotive domain as part of SCSs. Appearing problems that have been identified in the motivation are broken down in Section 1.2. The objectives of this dissertation are specified in this section as well. Subsequently, approaches including contributions for the problems and objectives are provided. Section 1.4 gives an overview over the whole thesis. Finally, a list of publications is given that has been published during the doctoral study.

1.1. Motivation

The invention of the automobile goes back to 1886. In that year, Carl Benz applies a patent for a vehicle with a combustion engine on January 29. [Fer86] The vehicle with the name *Benz Patent Motor Car*, that has been constructed by Carl Benz, was an innovative invention at that time since there was no construction composed of combustion engine, chassis and motor-drive mechanism at any time before [MB17]. Since the invention of the automobile in 1886 the development has been continuously moving forward. At the beginning the development of comfort equipments were at the focus. In the 1960s safety played an important role for the first time. [AE15] Today, it is mandatory to take safety requirements preventatively into consideration to guarantee a maximum degree of safety. This is an experience which the Ford Motor Company had to undergo. Thereby, in the late sixties the manufacturer had to recognise that it is essential to define mature safety goals. The former Chief Executive Officer (CEO) Lee Iacocca decided at that time to produce a fuel-efficient and inexpensive car. The car should weigh 2.000 pounds and the price should not exceed 2.000 $. The market launch of that car, named Ford Pinto, was in 1970, i.e. the schedule for the development was quite tight. Due to lack of time the CEO decided to neglect some safety checks. There was a hazard identified late in the design process that the fuel tank being damaged and catching fire after a rear-end crash. Notwithstanding the fact that the hazard is extremely life-threatening, the management decided not to make any changes on the tank position due to the time and cost factor. As a result there were 53 lethal accidents and many injuries. [Ord+09] This example shows that it is essential to define major safety goals which have to be accomplished in any circumstances.

In the last few years the trend is moving towards autonomous driving or comfortable driving. To provide these features it is necessary to supply the cars with a lot of assistance systems. These include, e.g. an Adaptive Cruise Control (ACC), a Lane Assist (LA) or a Lane Departure Prevention (LDP). [Ger+10] Each of them are cross-linked, both inside and outside the car, to communicate with other interfaces and Electronic Control Units (ECUs) or to detect other cars in the local area, i.e. in front or in the blind spot. For this purpose, numerous wired and wireless interfaces and ECUs are deployed to process data. In general, the term *wired interfaces* refers to bus systems like FlexRay, Controller Area Network (CAN) or Local Interconnect Network (LIN) that are plugged inside the car [KH10]. The term *wireless interfaces* concerns Bluetooth connections as well as radio waves that are sent and received by sensors. These Advanced Driver Assistance Systems (ADASs) and wireless interfaces unfortunately provide vulnerabilities. There is a distinction between active and passive hacking attacks. Passive hacking attacks are made in the consciousness of the driver or the vehicle owner. These typically include changing the mileage or increasing of the top speed. Active hacking attacks are made from third parties maliciously to cause damage in any way. In that way it is done without the driver's knowledge, i.e. the driver notices the damage at a late stage or at the moment of the damage. In 2015 there was a fatal hacking attack which has been called *Fiat Chrysler Hack*. Security researchers were able to take over control of some vehicles of the brands Jeep and Dodge. The researches could operate air-conditioning, braking and power transmission. The hack was performed by means of a notebook which was located 15 kilometres away from the car. They could send commands and access the CAN bus which is responsible for the cross-linking of ECUs in automotive vehicles. [Rin15] In another case Chinese security researchers could hack a driving Tesla Model S. They were able to open the tail gate and to trigger a braking operation from remote. For that purpose the driver must use an internet connection inside the car, e.g. if the driver is searching for a charging station. [Kli16] Both, active and passive hacking attacks, can endanger human life without consideration of safety aspects. Therefore, it is necessary to define safety goals to guarantee a maximum degree of safety.

Besides that, cars are developed further and thus the corresponding goals and systems of cars may change and can have some impacts on the resulting safety. Therefore, these impacts of the individual sub-systems are taken into account to provide high safety. The customised configuration of cars goes back to 1903 when Henry Ford presents the Model A. In those days, the customer could order an optional row of seats. Furthermore, the customer could decide whether the car roof consisted of rubber or leather. [Rei04] Moreover, it is usual nowadays to configure its own customised car with respect to equipment, motorisation and so on, i.e. each configuration has different Safety, Security and Timing (SST) requirements and implications. From an economic point of view, it is not possible to check each configuration with regard to SST since it would exceed cost limits.

1.2. Problems and Objectives

The first section of this chapter presented some serious problems in case that safety and security requirements are omitted. It is the aim of this thesis to circumvent these problems. However, to reach the goals of this thesis different challenges have to be accomplished. Therefore, this section identifies problems and objectives that have to be solved in this thesis.

Multi-Concerns and Trade-Offs

Problem 1. Safety problems are major problems that can endanger human life. For instance, rear-end collisions may endanger human life. Therefore, it is mandatory to guarantee a maximum degree of safety. To achieve this, further concerns must be considered. Security and timing has to be taken into account as well. Let us assume the ACC system of an automotive vehicle has to be as secure as possible. For that purpose an encryption algorithm with highest security requirements is implemented that encrypts the data to be processed as well as data communication. The data communication includes commands that are sent via the bus system. On the one hand, data that has been read via the sensor are sent via the bus communication. These data include, e.g. the distance to the vehicle driving ahead. On the other hand, a command is sent to the actuator based on these data, i.e. it contains commands to enable braking or accelerating. If the data are tapped, wrong commands may be executed. This situation presents a security property. Furthermore, it is possible to execute commands at a wrong time. This way, delayed braking could be the result. In this context, we talk about the timing concern. As a consequence, a crash could be unavoidable. Moreover, it is also possible to cause a crash due to highest encryption standards. If the data or data communication is encrypted with a strong encryption algorithm it also needs more time to decrypt data. As a consequence, commands will be processed when the braking already should have initiated. In that case an optimal trade-off must be found between secure en-/decryption and their execution in time. If this can be accomplished, the ACC system is acceptably safe. It is not enough to prevent an individual safety goal but a combination of Safety, Security and Timing goals. It is necessary to check whether these goals are contradictory to each other. Therefore, often a trade-off between the different goals has to be calculated that solves the problem with an appropriate compromise.

Objective 1. *It has to be possible to define necessary Safety, Security and Timing (SST) requirements, i.e. Multi-Concerns (MC) have to be modelled. Furthermore, risks of the individual SST requirements and thus MC need to be taken into account. In this context, it has to be possible to develop and model Counter Measures (CMs) to avoid or mitigate risks. Moreover, a set of alternative solutions is needed for which an optimal trade-off can be calculated. The analysis procedure which calculates trade-offs is called Multi-Criteria Decision Making (MCDM).*

Change Impacts

Problem 2. Already the ancient Greeks or more precisely Heraklit said "πάντα ῥεῖ" [Sei99], i.e. everything is always in motion. This applies to software and system development and in particular to SCSs. The system or software development is constantly progressed, e.g. by introducing new features. As a result changes of the whole architecture or software product are needed. In this context the risk and relative importance of SCSs is on change since amendments regarding safety, security or timing issues entail to update the risk assessment and relative importance. In practice, it is too laborious to perform a MCDM once again. Normally, just a small part differentiates from the original MCDM. Therefore, it is essential to find out which parts of the MCDM are concerned of the individual change modifications.

Objective 2. *A change impact analysis has to be developed to detect which parts of the MCDM process the individual amendments affect. Thereby, horizontal and vertical traceability has to be enabled to retrace the effect chain. In this context, it has to be differentiated between Best Case (BC) and Worst Case (WC) change impact analysis, i.e. there is a minimum or maximum number of concerned impacts. For each of them, BC and WC as well as for all dependencies within the entire process chain well-defined impact rules have to be established. In this way, a maximum degree of safety can be enabled with a minimum of effort.*

Software Product Lines

Problem 3. Nowadays, software based products are configured via a modular system, i.e. the customer can, e.g. build his or her own customised car from a series of possible configuration options to allow variability. For modern mid-range cars there are several millions of configuration options. However, each configuration, i.e. product, which can be ordered by the customer, has to fulfil certain SST requirements. In practice, it is not possible to take every Software Product Line (SPL) configuration into account to calculate an optimal trade-off which is acceptably safe and secure. Let us assume we can configure assistance systems for a car consisting of three items where it is possible to choose only one component, two components or the maximum configuration of all three items. Depending on the selection made regarding the three assistance systems different measuring devices (radar-, ultrasonic- and camera based sensors) with different SST standards will be deployed can be derived from an underlying system model. In this way, it has to be ascertained which system components are coherent and which are incoherent. Otherwise, this could lead to neglecting important SST requirements. In summary, it is not possible to perform a MCDM with all configuration items to get a maximum degree of safety. Therefore, an automated approach is needed to guarantee a maximum degree of safety and to reduce complexity and to save working time.

Objective 3. *It is the aim of this objective to model SPLs taken SST issues into account. These models have to be standardised, easy to understand and cover all variabilities of SPLs. Furthermore, these models need to linked to allow calculation of MC trade-offs, impacts and SPL based trade-offs. Moreover, it is an essential goal to cluster issues with similar SST requirements to reduce overhead and to enable reusability. As a result the MCDM of Objective 1 should be performed for individual selected SPLs.*

1.3. Approaches

Section 1.2 described problems and objectives of this thesis. The following paragraphs will give an abstract understanding of the MCDM, change impact analysis and complexity reduction of SPLs.

Multi-Concerns and Multi-Criteria Decision Making

Since SCSs may cause serious problems, it is first of all essential to identify potential failure modes, i.e. to identify the manner in which the failures of a SCS occur. These potential failure modes may arise when realising SST requirements. The starting point is a top-level safety goal to achieve a maximum degree of safety, i.e. the structure is hierarchically to break down goals as good as possible. This central safety goal is refined to smaller safety-, security- or timing goals which represent the SST requirements. If the safety-, security or timing goals are represented by axioms we speak about Point of Vulnerability (POV). Parallel a system is specified in a System Model (SM) regarding hardware and software specifications derived from the requirements. In the next step, alternative solutions, i.e. a set of safety-critical hardware and software configurations, are designed in accordance with the SM. Based on these alternative solutions a MCDM is performed, i.e. the algorithm of the MCDM decides which alternative solution best fulfils the SST requirements. All collected goals and POVs are transferred into an hierarchical model, the Safety-, Security-, and Timing Model (SSTM). The SSTM consists of two or more alternative solutions to calculate a significant trade-off. Furthermore, the Attack and Defence Tree (ADT) is suitable to specify Counter Measures (CMs) for security-critical issues of the SSTM. To assess risk of the SST goals and POVs the Failure Mode and Effects Analysis (FMEA) and Attack and Defence Tree Analysis (ADTA) are applied, whereas the ADTA is based on the ADT. If there are any risky goals or POVs the preceding steps are repeated until the risk assessment reaches a safe, secure or timing optimised level. To determine which of the SST goals within the SSTM are most important a pairwise comparison is performed. For this purpose the MCDM algorithm Analytic Hierarchy Process (AHP) is applied. The reason for applying the AHP is that the SSTM with its goals is built up hierarchically and the decision making of the AHP is based on an hierarchical approach, too. I.e., the necessary calculations do not require any scalings which might falsify the result of the MCDM since the AHP and Goal Structuring Notation (GSN) work always on the same hierarchical level. Data which are submitted by domain experts influence

the result of the MCDM. It is decided which alternative solution is applicable with highest accuracy. However, if the trade-off is not solvable, the AHP assessments are repeated until it is feasible. The whole abstract procedure is summarised in Figure 1.1. [LFB18]

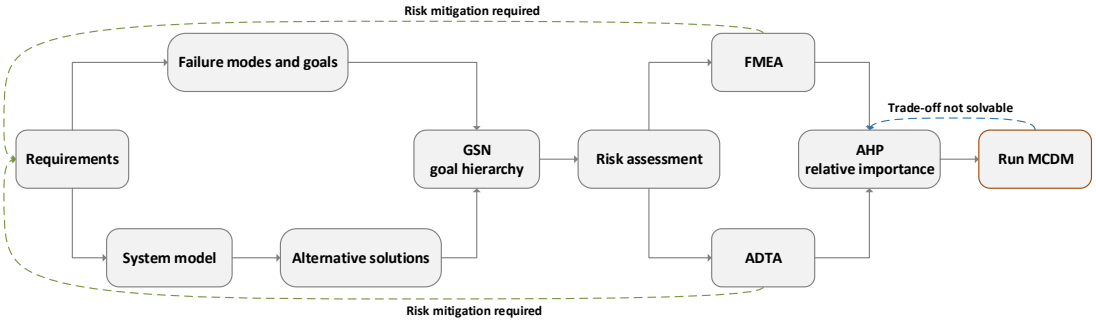


Figure 1.1.: Abstract procedure of the Multi-Criteria Decision Making [LFB18]

Change Impact Analysis

In general, the change impact analysis is initiated by stakeholders, e.g. safety experts or developers which make change requests. In this way, corresponding domain experts are involved. Subsequently, it is necessary to define change requests affecting the SM, SSTM and ADT. These change requests include, amongst other things, corresponding requirement changes and a triggering operation by which the change impact analysis is started. Afterwards, the change requests are linked with corresponding nodes of the above mentioned model types, namely SM, SSTM and ADT. Subsequently, it is distinguished between two dimensions: The first one is based on structural impact rules which concerns dependencies between the individual model nodes. Thereby, dependencies between SM and SSTM as well as dependencies between SSTM and ADT are considered. Moreover, dependencies between the individual model types (SM, SSTM and ADT) itself have to be taken into account. For instance, changing a goal within the SSTM may have effects on another goals within the SSTM. The second one determines the impact by means of calculation of attribute values for model elements. By selecting an impact operation, e.g. modifying, deleting or extending and dimensioning the impact analysis will be performed. The abstract procedure of the change impact analysis is summarised in Figure 1.2.

Multi-Concerns in Software Product Lines

Variability is an important aspect to realise SPLs and individualised products. In the context of MC we start with a Feature Model (FM) which considers functionality of a SCS. Furthermore, a SM, e.g. a component diagram is needed which covers all system components. Features of the FM and system components of the SM

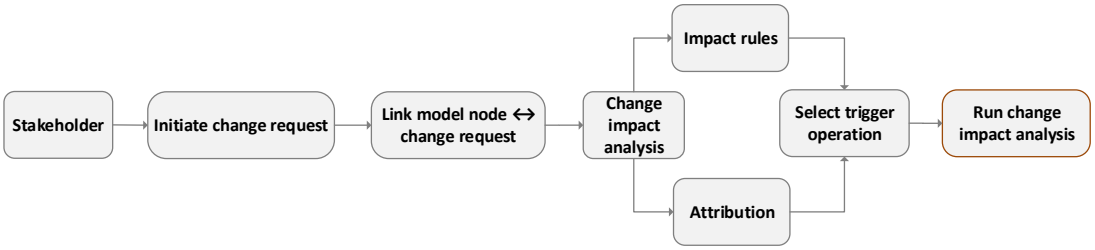


Figure 1.2.: Abstract procedure of the change impact analysis [LRB19]

are linked to detect indirect shared dependencies. For instance, the feature *LDP* may depend on *ACC* although *LDP* is not contained in the *ACC* SPL. Moreover, it is necessary to link the features of the FM with the goals or POV of SSTM to perform MCDM at the end. Afterwards, it is essential to annotate features of the FM with SST tags, i.e. the types of concerns (at least one concern) is assigned to the corresponding features. Subsequently, an attribution for each feature of the FM is defined to cluster semantically similar features and thus to reduce complexity. This step is essential, since it is not possible to analyse each SPL for reasons of time and costs. Therefore, we must find configurations with identical SST requirements. Since SPL approaches aim to customise products it is also necessary of this approach to select an individual configuration, i.e. one or more branches within the FM have to be chosen. Therefore, trade-offs can be calculated in consideration of individual SPLs. In summary, the SPL approach reduces the complexity of the SPLs and the MCDM based on reduced models is calculated. The procedure as described above is depicted in Figure 1.3.

Figure 1.3.: Abstract procedure of the SPL based MCDM [LB19]

Methodology at a Glance

The last sections described the single approaches of this thesis which can be merged to a coherent approach (cf. Figure 1.4). In general, there is a differentiation between functional and qualitative requirements. Functional requirements like *Keyless go* describe functionality of a software system whereas qualitative requirements like *Keyless go acceptably secure against manipulation* define criteria for quality of a software system [Poh10] [Som11]. The functional requirements are modelled by FM and SM to realise SPLs and technical systems in this thesis. To consider qualitative and safety-critical requirements a SSTM is needed to calculate optimal trade-offs afterwards. The methodical procedure and prerequisites of this thesis consists of seven essential steps:

1. Requirements are defined in a consistent and standardised format, e.g. in form of goals.

2. Functional requirements are implemented by creating or using an existing FM and SM.

3. Qualitative requirements are fulfil properties, e.g. maintainability or modularity and are restricted to SST are applied SSTM. The SSTM can be extended by an ADT to perform security analyses for individual security aspects.

4. There is a linking between the SSTM and an underlying SM to derive suitable alternative solutions.

5. The FM is linked with the underlying SM to get shared dependencies of SPLs.

6. FM and SSTM are linked to calculate the optimal trade-off for an individual SPL at the end.

7. Finally, the individual SPL configuration is transferred into the SSTM to determine the resulting trade-off.

When performing a change impact analysis, it is, analogous to requirements, differentiated between functional and qualitative change requests. The effects of the individual source models on the target models yield from the corresponding linkings or dependencies which can be taken from the description of the methodical procedure or Figure 1.4.

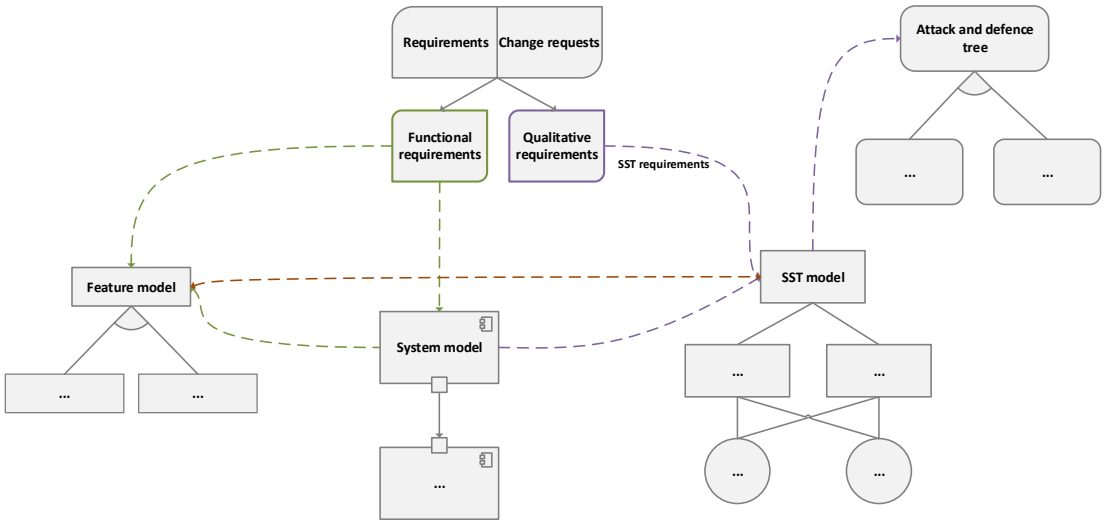


Figure 1.4.: Methodical procedure of the thesis

1.4. Outline

This section describes the logical structure of this thesis which is depicted in Figure 1.5. In general, the thesis consists of three parts: Basics, main part and evaluation. Each of them is subdivided into two or more separate chapters. All objectives and solutions are covered by the main part.

Chapter 1 Introduction exposes the topic of this thesis. First, a short motivation is presented to set the context. Subsequently, the problems and objectives are extracted and it is shown how they can be solved. These problems and objectives are solved by the corresponding approaches which are presented in an abstract manner. Finally, an outline of this thesis as well as a list of publications is given. These publications has been published during the doctoral study of this author.

Chapter 2 Foundations provides an introduction to Multi-Concerns. Moreover, techniques are described which are required for safety- and security analyses. Furthermore, the necessary modelling notations are introduced which are used in later parts of the thesis. Finally, MCDM algorithms are presented to calculate a trade-off.

Chapter 3 Multi-Concerns and Multi-Criteria Decision Making covers the first main chapter. In general, an approach is presented to model failure modes and goals in context of SCSs to calculate the best possible solutions. Thereby, risk assessment using the FMEA and ADTA play an important role. Furthermore, it is described how to use a MCDM by applying the AHP. In this context, two algorithms are presented, the Pairwise Comparison Mode (PCM) as well as the RPN Comparison Mode (RCM).

Chapter 4 Change Impact Analysis describes the second main chapter of the thesis. It is described how some changes, e.g. installation of new safety-critical components affect the calculation of optimal trade-offs. In this context, modifications of individual models in consideration of the SST properties are necessary and are described in detail.

Chapter 5 Multi-Concerns in Software Product Lines is the last chapter of the main part and covers the application of SPL taken MC into account. An approach is presented how to model SPLs by means of FMs which consider the SST concern. The complexity of safety-critical SPLs is reduced by means of semantic clustering. Subsequently, it is described how the reduced SPLs is applied for subsequent MCDM. Finally, the change impact analysis of Chapter 4 is extended.

Chapter 6 Realisation and Evaluation gives an overview of the implementation of the three main chapters which has been realised by an Eclipse EMF[1] and Sirius[2] plugin. Subsequently, it carries out an extensive scenario-based

[1] `https://www.eclipse.org/modeling/emf/`

[2] `https://www.eclipse.org/sirius/overview.html`

evaluation considering the three main chapters of this thesis, namely the Multi-Concerns and Multi-Criteria Decision Making, the Change Impact Analysis as well as the Multi-Concerns in Software Product Lines. Finally, two case studies show the applicability of the approach.

Chapter 7 Conclusion and Outlook summarises the thesis by recapitulating the problems, objectives and the approach. Furthermore, an outlook for future research works is given.

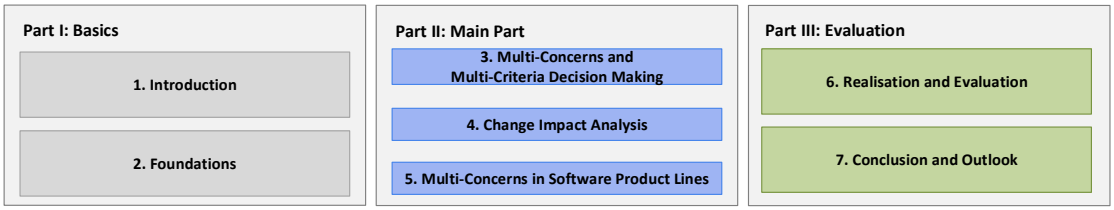


Figure 1.5.: Outline of the thesis

The three grey boxes of Figure 1.5 show the general reading order of this thesis. However, if the reader is familiar with the basics, which are presented in Chapter 2, he or she can continue reading with the main part of this thesis. As indicated by numbering of the main chapters it is highly recommended to start reading with Chapter 3 followed by Chapter 4 and Chapter 5 since it is the logical procedure of the approach and the thesis. The implementation details of Chapter 3 to Chapter 5 are realised by Chapter 6. Furthermore, the evaluation is described in Chapter 6 as well. It is strongly recommended to read Chapter 6 and Chapter 7 as a last resort.

1.5. Publications

During the PhD study of the author of this thesis some scientific publications have been published. Applied concepts and results of the author, which have already been published or to which the author has been contributed, are cited at the beginning of the corresponding chapters. These also include supervised master theses. The following list gives an overview of all that publications with a brief description of contents.

1. Philipp Lohmüller, Andrea Fendt, and Bernhard Bauer: "Multi-Concerns Engineering for Safety-Critical Systems". In: *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development Volume 1: MODELSWARD*. Funchal (Portugal), 2018, pp. 504-510 [LFB18]
   The author of this thesis, together with an associate researcher, presents a concept which is essential for calculating trade-offs by considering MC. The concepts and ideas have been developed in cooperation with a master student within the scope of a master thesis [Fen16]. The master student has been supervised by the author of this thesis. The publication itself has been

written down by the author of this thesis. This work is primary essential for Chapter 3.

2. Philipp Lohmüller, Julia Rauscher, and Bernhard Bauer: "Failure and Change Impact Analysis for Safety-Critical Systems". In: *Business Modeling and Software Design*. Lisbon (Portugal), 2019, pp. 47-63 [LRB19]
   The author of this thesis, together with an associate researcher, presents a concept which is essential for calculating failure and change impacts by considering SST concerns. The concepts and ideas have been developed in equal shares by the authors. Furthermore, this paper "has been partially supported by the German Federal Ministry of Economics and Technology (BMWi) in the framework of the Central Innovation Program SME (Zentrales Innovationsprogramm Mittelstand) within the project CBMD[3] [LRB19]". This work is a central part of the second chapter.

3. Philipp Lohmüller, and Bernhard Bauer: "Software Product Line Engineering for Safety-critical Systems". In: *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development Volume 1: MODELSWARD*. Prague (Czech Republic), 2019, pp. 211-218 [LB19]
   In general, the authors describe in their paper how the complexity of SPLs in context of SCSs can be reduced to analyse manageable trade-offs. The content of this paper is a basic for the third main chapter. In this paper, the first author developed the concept and main ideas of the research questions.

[3] `https://www.informatik.uni-augsburg.de/en/chairs/swt/ds/projects/mde/cbmd/`

2 Foundations

The current chapter covers the basics which are necessary to understand the following chapters in detail. First, an introduction to MC is given. Subsequently, Section 2.2 describes the foundations for risk management in the area of safety and security. Furthermore, necessary model notations are introduced in Section 2.3 which are applied in the main part of this thesis. These include the Goal Structuring Notation (GSN), Feature Models (FMs) and System Models (SMs). Moreover, the thesis covers Multi-Criteria Decision Making (MCDM). Therefore, the necessary basics are explained within this chapter. These include, inter alia, the AHP which is essential for calculating trade-offs in later course of this thesis.

2.1. Multi-Concerns

This chapter aims to introduce the most common concerns which are taken into account. These include safety, security as well as the timing concern. There is still no widely accepted definition for concerns. Even though there is an intuitive understanding of concerns:

Definition 2.1 (Concern)**.** A concern defines one or more properties and requirements for a system which need to be fulfilled to comply with quality attributes as good as possible [RBC05].

In this dissertation it is the primary goal to achieve safety. However, to accomplish a maximum degree of safety, the security and timing concern must be considered as well. The term MC defines the correlations between the individual concerns. Figure 2.1 gives a first overview of the relationships. Hereinafter, for each dependency between the individual concerns an obvious example is given:

- **Safety** $\rightarrow$ **Security:** Sensor data of individual ADTAs are transmitted in encrypted form. Otherwise wrong actions could be done.
- **Security** $\rightarrow$ **Safety:** If an insecure encryption algorithm has been chosen for encrypting data traffic of an ACC cyber attacks are possible. In the worst case, the braking process could be prevented.
- **Safety** $\rightarrow$ **Timing:** Each automotive vehicle, which has been equipped with an Emergency Brake Assist (EBA), has to be able to react within fractions of a second.

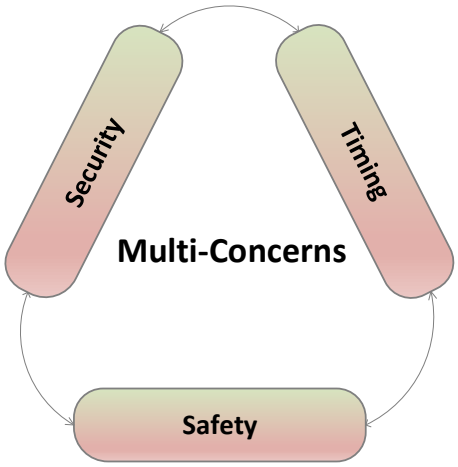


Figure 2.1.: Multi-Concerns: Dependencies between Safety, Security and Timing

- **Timing → Safety:** It is a legal requirement that every registered car has to be equipped with an airbag. However, if the airbag does not trigger in time it poses a hazardous risk for all occupants.
- **Security → Timing:** Let us assume our car is equipped with an ACC. It is mandatory to encrypt data traffic to protect against cyber attacks. However, if the encryption algorithm is more secure than required it takes too much time to decrypt corresponding commands.
- **Timing → Security:** Under the assumption that the ACC is working correctly, an upper time limit needs be met for individual commands, e.g. accelerating or decelerating. Therefore, a suitable encryption algorithm must be elected with respect to upper time limit in order to protect data (traffic) as secure as possible.

2.1.1. Safety

The concern safety has been used in context of launching safety belts to guarantee safety for road users [Eva86]. For the term safety there is a large number of definitions. In conclusion, the term safety is defined as follows:

Definition 2.2 (Safety)**.** Safety as defined in the *IEC 61508* standard [IEC97] "is the detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the consequence of the hazardous event [IEC97]." Furthermore, it describes a state that a person or a human being is protected against any harms or unwanted side effects [RM90].

Hence, the safety property is violated if the driver is in danger due to lack of safety protection. Once, if the driver or other occupants neglect the obligatory wearing

of seat belts the safety of all the human beings inside the car are endangered since a traffic accident could occur at any time whether the driver is inattentive or other road users may bear the guilt. In general, safety not only depends on behaviour of drivers or occupants.

Definition 2.3 (Safety-Critical System)**.** "A Safety-Critical System is a system where human safety is dependent upon the correct operation of the system." [ST16] According to [Som18], malfunction of the system may entail death, personal injury, environmental damage or (partial-)damage to the system.

In the context of safety there is a widely used norm for the development of Electric, Elctronic and Programmable Electronic Systems (E/E/PE) that consider safety functions. This norm is called *IEC 61508*. In general, the norm describes how safety-critical products have to be developed to mitigate potential risks. It is the aim of this norm that no hazards endanger operators and environment. Independent certification authorities like the German Technischer Überwachungsverein (TÜV) require a certification due to product liability law. In Germany, according to paragraph §4 ProdHaftG manufacturers are liable for failures and risks of suppliers. In the product liability litigation manufacturers demonstrate that proven techniques has been applied for risk assessment and safe product development. The *IEC 61508* norm also provides a life-cycle model, development cycles, a product architecture, an organisation structure as well as well defined documentations for third parties. [IEC97]

There are also safety aspects which concern the ADASs or the technique which is plugged inside and outside the car. Let us assume the driver is cruising along the highway with activated ACC, i.e. the driver anticipates that accelerating and decelerating are taken by the car itself without operating any pedals. However, if electronics of the ACC refuses suddenly, it also constitutes a significant safety risk, since a necessary brake process could be initiated not at all or too late. This process may endanger human life in a bad way. [Nil96] To prevent such accidents it is mandatory to perform hazard and risk analyses during the design and development process. Some of the commonly used hazard and risk assessment procedures are captured in Section 2.2.1 with a detailed explanation.

2.1.2. Security

Since automotive vehicles are provided with diverse ADASs and computer-aided information systems it is possible to manipulate the electronics of such vehicles. As already mentioned in Section 1.1 there is a distinction between active and passive points of attacks. Both, active and passive points of attacks, may endanger safety of road users and thus it is mandatory to counteract them. The term *security* has to be separated from *safety* and is defined as follows:

Definition 2.4 (Security)**.** Security defines to which extent a computer-based system is protected against data manipulation, data theft and unauthorised access [McG06] [IEC05].

There is a widespread norm for security, the *IEC 27001* which concerns in large parts to keep information assets secure and to avoid hacking attacks. Usually, such hacking attacks can be mitigated by applying a secure encryption algorithm which is used in a synchronous or asynchronous manner. Let us quote the ACC example from the previous section 2.1.1. The driver is assuming that the ACC works without any lacks. In the background hackers got access to the corresponding ECUs of the car, e.g. via wireless interfaces like Bluetooth. After getting access to the corresponding ECU the hackers can manipulate the system in an arbitrary manner. One possible scenario is to disable the braking function of the ACC system. A life-threatening rear-end collision is almost unavoidable. You can counteract this scenario by applying security procedures, e.g. suitable encryption modes. Besides the classical hacking attack there are more dangerous security issues. In this context a Denial of Service (DOS) attack poses a risk to the environment. Thereby, it may occur that some services or functions, e.g. measurement of the distance to the vehicle driving ahead are deactivated. [FGS05] A further important security aspect is the protection of privacy and data manipulation. As a result it may be possible to spy out confidential data, e.g. home address or phone number. Moreover, technical data of the car may be spied out. For the reason of distortion of competition it would be unacceptable. [FIM10]

As stated in the introduction of this chapter or in Figure 2.1 better security boosts the safety as well. Analogous to safety there are some preventative methods to promote security. These methods will be explained in Section 2.2.2 in an extensive manner.

2.1.3. Timing

The last two concerns covered the safety and security concerns. However, there is also a further factor that influence safety in a considerable manner. In that context, the observance of upper and lower time limits is of considerable importance. Therefore, we can define timing as follows:

Definition 2.5 (Timing)**.** The term timing defines that certain tasks have to be performed within a given time interval [Per+12].

In practice, timing is the response times or Worst Case Execution Time (WCET) analyses, i.e. the upper time limit for executing a task is pre-determined. In some cases the lower time limit must be considered as well. This is the case if some steps could be performed only then if other calculations or processes have been done. [BCP02]

Let us recall the ACC example from the previous two concerns. We already made it more safe and secure by applying a secure encryption algorithm. However, this may constitute a serious safety risk if commands, that contain essential actions, e.g. accelerating or decelerating, are received not in time. This may cause an accident and thus it endangers safety of inmates and other road users.

In summary, there has to be an optimal trade-off between the three three concerns Safety, Security and Timing with the primary goal to achieve a maximum degree of safety. The techniques and approaches, that we need therefore, are explained in detail hereinafter. [Ans+11]

2.1.4. Influences of Security and Timing on Safety

In Section 2.1 it has been shown that all three concerns, i.e. SST influence each other. In this section, we will go into detail on how security and timing influence safety and why safety is our primary goal. In general, safety is responsible for maintaining human life, therefore safety plays an important role in MC engineering.

For the sake of completeness, besides security and timing there are further influences, e.g. economic influences, political and personal interests. However, these influences are no main part of this thesis and will be therefore not further explained. In practice, some concerns may contradict to each other because the individual concerns only make sure that safety is guaranteed by applying its own appropriate measures. For instance, in the near future automotive vehicles will drive autonomously. In this context, it will be necessary to enable car-2-x communication, i.e. communication with traffic lights needs to be accessible as well. [Sta+11] However, to ensure that no third parties manipulate the commands of acceleration process and decelerating process a secure encryption is required. By applying an encryption algorithm of highest security level it may happen that the commands are not be executed in real-time. In that way, it would be possible that the car gets the command for accelerating although the traffic light shows already the red light. This example shows that security influences timing and thereby safety.

The last example showed that security influence timing and thereby endanger safety. It may also be possible that security violates safety aspects. The following example from the past demonstrates this scenario in more detail. Due to the terrorist attacks of September 11th, 2001 on the World Trade Center in New York City it has been decided to perform stricter security measures. Given the fact that the air-planes from the terrorist attacks had been hijacked a security measure has been introduced that the access to the cockpit is denied during the flight. This security precaution became an obstacle for the Germanwings flight 4U9525 on March 24th, 2015 that crashed in the French Alps. The captain within the cockpit has full control over the door and can even inhibit emergency access. Based

on voice records, the co-pilot is suspected of deliberately destroying the plane while preventing the captain from re-entering the cockpit. I.e., increasing security against hijackers intensified reliance on the pilot being left on his one and carrying full responsibility for flying the air-plane [Yu15] In this case, the stricter security measure influenced safety in a negative way. In most cases, timing influences safety in a direct manner. This is due to the fact that timing issues often depend on security aspects as the car-2-x example has been shown. The negative influences of MC in direct manner or indirect manner, that has been reflected in this sub-section, should be solved by means of the approach that are presented in further chapters of this thesis.

2.2. Risk Management

In Section 2.1.4 a series of hazards and risks have been presented which should be circumvented by applying correct system engineering process. In general, the concept of system architecture is made by a systematic approach which considers design of the entire system. In this way, it is ensured to avoid unusable measures and to apply measures which fulfil safety or security requirements of the system. In this context, costs will be saved in long-term vision. As depicted in Figure 2.2 the systematic approach consists of four sequential phases which are explained in more detail hereinafter:

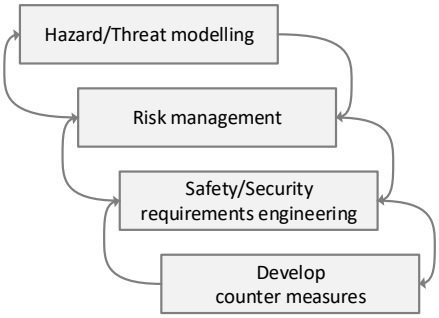


Figure 2.2.: System engineering process [MLY05]

- **Threat modelling:** According to [Sho14] a system is modelled methodically from perspective of hazard or threat source. In this context, the following three steps are performed:
  1. Description and specification of the system, e.g. by means of a SM.
  2. Vulnerabilities and access points to the system.
  3. Identification and analysis of the hazards or threats. In this way, safety-critical system components will be revealed.
- **Risk management:** For the individual threats or hazards risk management is mandatory. According to [MLY05] it is subdivided into two steps:

1. Risk assessment, i.e. a risk analysis is performed either qualitatively or quantitatively.
2. Classification of the threats or hazards, i.e. it is decided whether the risk is acceptably safe or secure.

- **Safety/Security requirements engineering:** It is the aim of the requirements engineering process to define safety or security requirements. This process consists of four steps which are explained in more detail in Section 3.3: *Identifying, analysing, specifying* and *validating* requirements. [Som11]

- **Develop Counter Measures:** CMs are necessary to fulfil safety or security requirements and to mitigate risks of threats or hazards. Applying a CM requires the usual phases of software development cycle: design, implementation, testing and maintainability. [MLY05]

2.2.1. Safety Risk Management

As indicated in Section 2.2 violation of safety aspects can endanger human life primarily. To circumvent this, some preventative risk management methods are necessary. It has to be pointed out that negligence of safety can cause financial and economic losses. In this way, by means of preventative risk assessment direct follow-up costs of personal injuries for, e.g. medical care, salaries or compensation payments. Furthermore, indirect follow-up costs of injuries can be avoided. These include financial penalties due to non-compliance of regulations or repair costs. Moreover, productivity can be boosted. If there is a recall campaign based on safety lacks it causes unnecessary costs. Such image loss can also influence global competitive ability and the associated export potential. [SAG18] Therefore, this section introduces three techniques to assess and mitigate potential safety risks.

2.2.1.1. Failure Mode and Effects Analysis

As already indicated in Section 1.1 there were several deadly accidents due to lack of risk assessment. Therefore, it is content and purpose of the FMEA to mitigate those risks in context of SCSs as much as possible by applying the FMEA. The FMEA was originally developed in 1949 as a military instruction titled *"MIL-P-1629 - Procedures for Performing a Failure Mode, Effects and Criticality Analysis"* [DOD49]. In the 1960ies the FMEA has been used for planning nuclear power plants. Furthermore, it has been used in avionics. Between 1970 and 1980 the FMEA has been used in automotive industry for the first time, namely from the Ford Motor Company in USA. The FMEA has been developed over the time and is now in use for development of electronics and software. In 2006 the standardisation to DIN EN 60812 has been specified. [Ber+09] It is the aim of the FMEA to prevent failures and to detect failures. For the failure prevention it is essential to indicate and to prevent failures in early stages of product cycle. The later a failure is indicated the more expensive the development costs are. The costs

will increase about 10 times for each later stage [Ber+09]. The failure detection has four essential goals:

1. Detection of possible fault sources that can cause (subsequent) failures. The detection is, e.g. possible by applying the Fault Tree Analysiss (FTAs) (cf. Section 2.2.1.2).

2. All causes and consequences need to be correctly identified, mitigated or avoided.

3. Faultless organisation of processes during the development cycle.

4. Vulnerabilities of the system, products or processes have to be identified so that a constructive revision can be performed.

To prevent and detect failures it is necessary to assess potential risk by means of the Risk Priority Number (RPN). It is useful to answer the following questions beforehand to calculate the RPN afterwards:

Definition 2.6 (List of Questions for Determining the RPN)**.** To determine the exact values of the RPNs a list of questions serves as an indication:

1. What can go wrong?
   a) What failures did occur in the past?
   b) What can go wrong within the sub-processes or the system's parts?
2. Why did this failure occur?
   a) What are potential causes of the failure?
   b) Are there any similar failures in the past?
3. What would be the impact of the identified failure?
   a) What would be the consequences?
   b) What could happen in a worst-case scenario?

The following four paragraphs provide insight into determining the individual partial calculations of the RPN. Furthermore, it is explained how to interpret and analyse the individual partial values.

Risk Priority Number

Definition 2.7 (Risk Priority Number)**.** According to [BMI17] the Risk Priority Number (RPN) is defined as follows:

$$RPN = Occurrence(O) \times Severity(S) \times Detection(D)$$

where

- O complies with the probability that a hazard occurs,
- S corresponds to the severity of a hazard and
- D complies with the probability that a hazard will be detected.

As already mentioned in Definition 2.7 the $RPN = O \times S \times D$. Thereby, each factor can range between 1 and 10. Conversely, this entails that the RPN can range between 1 and 1000. In general: The lower the RPN the better the potential risk. Depending on the value of the RPN the degree of risk can be identified. [Ber+09] Table 2.1 gives an overview whether CMs are required regarding the values of the RPN. If the RPN is lower than 125, the risk is acceptable or low that no substantial CMs are required. However, if the RPN is greater or equal 125 expansive CMs are needed. Moreover, it should be considered that applying CMs due to potential risk may yield new risk. Let us assume an automotive group has been researched by means of the RPN that an encryption algorithm of 128 bit is not strong enough for an innovative ADAS to protect against third parties. Therefore, they decided to use an encryption algorithm of 256 bit instead. As a consequence, the RPN is lower than 125 and thus acceptable. It has not been considered that the calculations of the functionality need much more time and thus poses a new risk for the occupants of the car.

RPN	**Risk of error**	**Counter Measures**
RPN = 1	None	No CMs required
$2 \leq$ RPN ≤ 50	Acceptable	Additional warning required
$50 \leq$ RPN ≤ 250	Medium	Additional protective CMs required
$250 \leq$ RPN ≤ 1000	High	Constructive CMs absolutely required

Table 2.1.: Interpretation of RPN [BMI17]

Hereinafter, the individual factors for calculating the RPN will be clarified in detail. These include occurrence, severity and detection. For this purpose Definition 2.8 to Definition 2.10 explain the factors of the individual formulas. The accompanying value ranges are specified in Table 2.3 to Table 2.7. As already mentioned in the preceding paragraph each of the three factors can take a value range between 1 and 10. Therefore, it can be concluded that the better the occurrence, severity or detection the better the resulting RPN.

Occurrence

Definition 2.8 (Occurrence)**.** [GJ15] defines *occurrence* as follows:

$$O = (f \cdot r) + v$$

where

- f complies with the probability of failure,
- r corresponds to the stay in area of risk and
- v corresponds to the vulnerability of risk.

f	**Probability of failure**
1	Misbehaviour is rarely expected
2	Misbehaviour is expected with a moderate frequency
3	Misbehaviour is expected very frequently
r	**Stay in area of risk**
1	Stay in area of risk rarely
2	Stay in area of risk only sometimes
3	Permanent stay in area of risk
v	**Vulnerability of risk**
0	Not vulnerable
1	(Very) vulnerable

Table 2.2.: RPN: Calculation of occurrence [GJ15]

General assessment criteria	**Frequency**	**Rating values**
High It is almost certain that failures occur on a large scale	1/10 1/20	10 9
Moderate Comparable to earlier production processes that leads to failures often	1/50 1/100	8 7
Low Comparable to earlier production processes that leads to failures occasionally	1/200 1/500 1/1000	6 5 4
Very low The process is controlled statically	1/2000 1/20.000	3 2
Unlikely The process capability is ensured	≈ 0	1

Table 2.3.: RPN: Assessment criteria of occurrence [GJ15]

Severity

Definition 2.9 (Severity)**.** [GJ15] defines *severity* as follows:

$$S = (i \cdot d) + r$$

where

- i complies with the degree of injury,
- d corresponds to the duration of damage and
- r complies with the chances of rescue and damage limitation.

i	**Degree of injury**
1	Minor injury
2	Moderate injury
3	Very serious injury
d	**Duration of damage**
1	No long-term damages
2	Tolerable long-term damages
3	Serious long-term damages
r	**Chances of rescue and damage limitation**
0	Good chances of rescue
1	Unfavourable conditions for rescue and damage limitation

Table 2.4.: RPN: Calculation of severity [GJ15]

General assessment criteria	**Rating values**
A very grave failure occurs that affects and/or regulatory compliance.	10 9
A grave failure occurs that causes discontentment with the customer. Safety aspects or regulatory compliance are not considered.	8 7
A moderate failure occurs that causes discontent with the customer. The customers will notice the impairment.	6 5 4
The failure is insignificant and the customer is interfered slightly. The customer just notices marginal impairments .	3 2
It is unlikely that the failure has noticeable impacts on the investigation object. The customer probably does not notice the failure.	1

Table 2.5.: RPN: Assessment criteria of severity [GJ15]

Detection

Definition 2.10 (Detection)**.** [GJ15] defines *detection* as follows:

$$D = (q \cdot c) + r$$

where

- q corresponds to the qualification of the endangered person,
- c complies with the complexity of the hazard and
- r corresponds to the changes of reaction

q	**Qualification of the endangered person**
1	Expert
2	Instructed person
3	Amateur, no instructed person
c	**Complexity of the hazard**
1	Low complexity, situation is transparent
2	Average value of complexity, situation is still transparent
3	High complexity, situation is hardly transparent
r	**Chances of reaction**
0	Good chances of reaction
1	Poor chances of reaction

Table 2.6.: RPN: Calculation of detection [GJ15]

General assessment criteria	**Frequency**	**Rating values**
Unlikely Hidden failure that is not discovered in the production or assembly	< 90 %	10
Very low Defect is more difficult detectable	> 90 %	9
Low Defect is easily detectable	> 98 %	6-8
Moderate There is a obvious defect. Handling of the defect by very precise (100 %) check possible.	> 99,7 %	2-5
High Functional defect that is discovered in the following work stages	> 99,99 %	1

Table 2.7.: RPN: Assessment criteria of detection [GJ15]

There are some further FMEA extensions that are explained in a short way:

- **Failure Mode, Effects and Critically Analysis (FMECA):** The necessity of the additional term *criticality* is due to the fact that the US army developed its own FMEA standard in which the RPN has been replaced by criticality levels. Instead of probability of occurrence, a failure rate is used. The likelihood of detection is omitted as well. [Ber+09]
- **Failure Mode, Effects and Diagnostic Analysis (FMEDA):** The FMEDA extends the FMEA with the property to analyse and determine diagnostics options for detection of dangerous failures [LPP11].
- **Design Review Based on Failure Mode (DRBFM):** The car manufacturer Toyota sets high quality standards with respect to new development and applies the FMEA. Unfortunately, the quality standards are violated if modifications to products, product requirements or production processes are made. Toyota determined that such modifications were the cause for risks and failures. Therefore, it is the aim of DRBFM to find failures that result from intentional modifications as well as unintentional modifications on the product or the process. [SIN03]
- ***X*-FMEA:** *X* stands either for System, Hardware, Software or Process. It is purpose to analyse risks of the corresponding areas by means of FMEA. [Ber+09]

2.2.1.2. Fault Tree Analysis

Besides the FMEA there are further risk assessment techniques which are presented in this and the following section: The Fault Tree Analysis (FTA) and the Hazard and Operability Study (HAZOP). Analogous to the FMEA these approaches are used for preventative risk and failure mitigation in context of safety-critical systems.

Similar to the FMEA some accidents happened due to lack of risk assessment but up to now it was not possible to detect potential causes of failure to avoid them in future. The FTA was developed in the 1960ies by H. A. Watson in New Jersey. Originally, it has been used in the avionics for air-planes from Boeing. In the 1970ies and 1980ies nuclear power stations were planned by means of the FTA. Furthermore, in this period evaluation algorithms of the FTA has been developed. In the 1990ies it was used in the automotive industry for the first time. Moreover, software tools for developing and evaluating FTAs were developed in the 1990ies. Nowadays, the FTA is used in all industry sectors. Such as the FMEA the FTA is already specified to the DIN standard. It is the aim of the FTA for safety-critical systems to model a system most realistic and to evaluate it. In this context the failure types and failure causes are detected. By means of the FTA functional relations between the individual failures are established. Moreover, it is the purpose of the FTA to describe the impacts of the failures on the system. In general, the FTA is an

approach guarantee quality preventively and to analyse the system. [Ber+09]

A fault tree is created top-down, i.e. it starts with the failure cause, e.g. that a car does not start (see Figure 2.3). Subsequently, this top-event is connected with boolean operators like *AND, OR, XOR*. We call these elements *gates*. This gate is now connected with sub-events, i.e. failures that can occur and may lead to the top-event. I.e., the car does not start. In Figure 2.3 the sub-events are *Controls status OR Electronic fault*. If the controls are not in proper position, the problem is that the car is in park position *AND* the foot is on the brake. For case of an electronic fault there are two options: The starter is faulty *OR* the wire is broken. [Sch15] The complete syntax of FTAs is illustrated in [MB87].

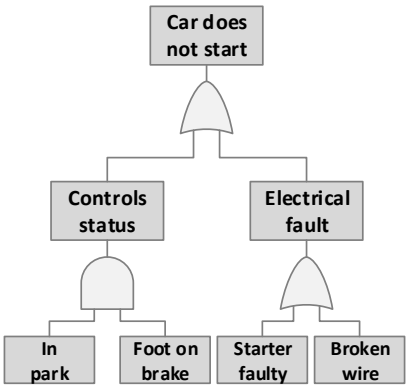


Figure 2.3.: Exemplary FTA: *Car does not start* [Sch15]

The evaluation is performed by means of stochastic probability calculations. All the (sub-)events are assigned probabilities that the failure occurs without any relationships between other events. The boolean operators are assigned mathematical operators like + (OR) or * (AND). In this way, the so called *critical path* can be calculated. This path symbolises the path from the leaf or leaves to the top event that leads to failure of the top event with highest probability. [Ber+09] Since the critical path is based on stochastic probabilities, it can be used for calculating the severity of the RPN in context of the FMEA [But07].

2.2.1.3. HAZOP

The HAZOP is a procedure for analysing hazards which is described in the IEC 61882 norm [IEC16]. It is the aim of the HAZOP to identify hazards systematically. In general, it addresses hazards which are triggered by systems and parts of them that do not work as expected or specified. The term *systems* include products, processes or software applications whereas *parts* include components or processing steps. [IEC16] normalises the HAZOP and describes the principles, application scope, procedure as well as audits. In general, HAZOP consists of four steps:

1. **Planning:** In the first step, the project manager defines the system which needs to be checked and the objective of the analysis. Furthermore, it is task of the project manager to put a team together including a head of HAZOP examination, system architect and a system operator.
2. **Preparation:** During the preparation phase the head of HAZOP examination collects some information about the system and prepares these data in terms of flow charts. Furthermore, he or she organises necessary meetings and modifies the list of guide words (cf. next step).
3. **Examination:** First, the objectives and the system are presented. Subsequently, the system is decomposed in its parts. For each part, its purpose is identified. It is mandatory to define all necessary elements, e.g. characteristics of materials or parameter. For each element it is essential to use each guide word to check whether a deviation is possible and may cause a hazard. An exemplary usage of the guide words is listed in Table 2.8.
4. **Evaluation of the results:** By means of a table consisting of the columns:
   - (sub-)systems
   - parts
   - parameter
   - guide word
   - possible causes
   - possible consequences
   - possible CMs

   The results are documented and analysed. The significant findings should be documented separately in a second risk table that consists all essential risks additionally. [IEC16]

The HAZOP analyses causes AND consequences of hazards whereas the FMEA only analyses the effects (by means of the RPN) based on causes and the FTA vice versa. However, the FMEA and FTA has been established as standard risk analysis method. Nonetheless, the HAZOP is used as an optional additional risk assessment to the FMEA and FTA. Furthermore, the HAZOP is a systematic approach which cannot prove that all hazards have been considered. Moreover, the HAZOP requires an experienced head of examination and a concrete and complete documentation of the system.

2.2.2. Security Risk Management

As already mentioned in Section 2.1.4 security may influence safety. For this reason it is mandatory to consider security risk management as well. In the same way as safety risk management neglecting security risk management may also cause long-term financial expenses and costs. Hereinafter, some graphical and structural approaches are presented to mitigate security risk.

Deviation type	Guide word	Example interpretation for a E/E/PE
Negative	NO	No data or control sign passed
Quantitative modification	MORE	Data passed at higher rate than intended
	LESS	Data passed at lower rate than intended
Qualitative modification	AS WELL AS	Some additional or spurious signal present
	PART OF	Data or control signals are incomplete
Substitution	REVERSE	Normally not relevant
	OTHER THAN	Data or control signals are incorrect
Time	EARLY	Signals too early with ref. to clock time
	LATE	Signals too late with ref. to clock time
Order or sequence	BEFORE	Sig. earlier than intended within sequence
	AFTER	Sig. later than intended within sequence

Table 2.8.: HAZOP: Examples of guide words [IEC16]

2.2.2.1. Attack and Defence Tree

In general, an ADT consists of two parts, *the attack tree* and *defence tree*. The attack tree is an acyclic directed graph, by which from attacker's perspective it is analysed, which vulnerabilities can be exploited if a threat is realised. In this context, all related attacks are modelled. Each of them represents an alternative way to realise the threat. The latter is represented by means of the root node and describes the main goal of an attacker. This goal is refined conjunctively or disjunctively by means of sub-goals using *AND* or *OR* nodes. The refinement is performed recursively as long as the refined sub-goal represents an elementary action of the attacker. To apply an attack the realisation of several elementary actions are necessary. The super-ordinated goals decide the kinds of actions. To realise an *AND* node it is mandatory to apply all sub-goals refined by them whereas for the realisation of an *OR* node just one of them has to be fulfilled. Attack trees can be transformed into Disjunctive Normal Form (DNF). By applying the DNF calculations of the analysis can be more easier and analysis results are more manageable. If an attack tree takes on large dimensions it will be more easier to find individual attacks by means of the DNF structure. Attack trees enable the basic form of statistical threat analysis and can be extended by different ways. For instance, leaf nodes may be attached with attributes for further calculations. These attributes will be passed up to the root node. In this way, each node is annotated with this attribute. Examples for boolean attributes may be like possible/impossible, easy/difficult, cheap/expensive, legal/illegal or special equipment mandatory/not mandatory. Besides of boolean expressions it is possible to assign continuous values, e.g. costs, success probability of the attack or the probability that an attacker performs an attack. [MO06]

Applying attack trees only enables an analysis of threats from attacker's perspective. Thereby, it is not taken into account which CMs mitigate risk for realising a threat efficiently and economically. To enable this scenario attack trees are ex-

tended by defence trees while adding defender's perspective. We call the extended attack tree ADT. For this purpose, each leaf node in the attack tree is extended by a set of nodes which corresponds to CMs and mitigate the risk of the threat. Similar to the attack tree, the extended ADT also enables analyses. [BFP06]

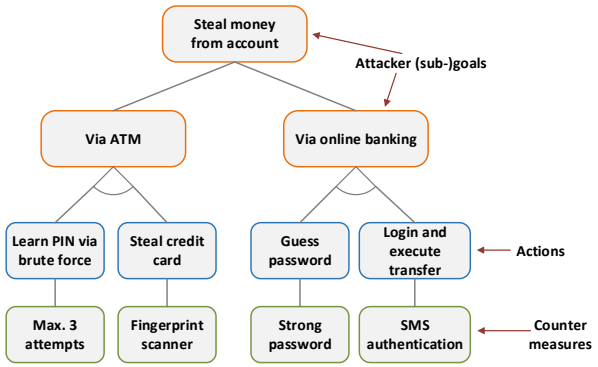


Figure 2.4.: Exemplary ADT: *Steal money from account* [KW18]

Figure 2.4 shows an exemplary ADT which covers the threat *Steal money from account*. In general, the orange bordered nodes show the attacker (sub-)goals, the blue bordered nodes the actions and the green bordered nodes the CMs. The root node *Steal money from account* is refined into two sub-goals *Via ATM* and *Via online banking*. In this case, the attack can be performed by only one of them. If the attacker makes a withdrawal via the ATM, he or she has to learn the Personal Identity Number (PIN) *AND* steal the credit card. Both actions have to be successful for the attacker to get money. It can be counteracted with a maximum number of three attempts or a fingerprint scanner. Otherwise, if the attacker catches the money via online banking the password has to be guessed *AND* the money transfer must be executed. Allocating a strong password or using a SMS authentication counteracts the threat. [KW18]

2.2.2.2. Extended Influence Diagram

Besides the ADT there is another technique for security system engineering: The Extended Influence Diagram (EID). Such as the ADT the EID provides a graphical representation for modelling uncertain variables and decision problems and enhances the Bayesian networks. [SEJ08]. The necessary foundations of Bayesian networks has been described in [Nea03] in detail. In general, the EID consists of three node types: *Utility node*, *decision node* and *chance node*. The decision nodes, which are depicted as rectangle, represents possible decision alternatives. The utility node is represented as rhombus and provides the outcome of a decision node which is quantitatively assessed. The chance nodes, which are depicted as ovals, consist of states from a finite domain. The relationships represent causal dependencies and interplay between the utility node, chance nodes and chance nodes. Finally, the EID provide calculations of quality attributes by means of probability values. [ES09]

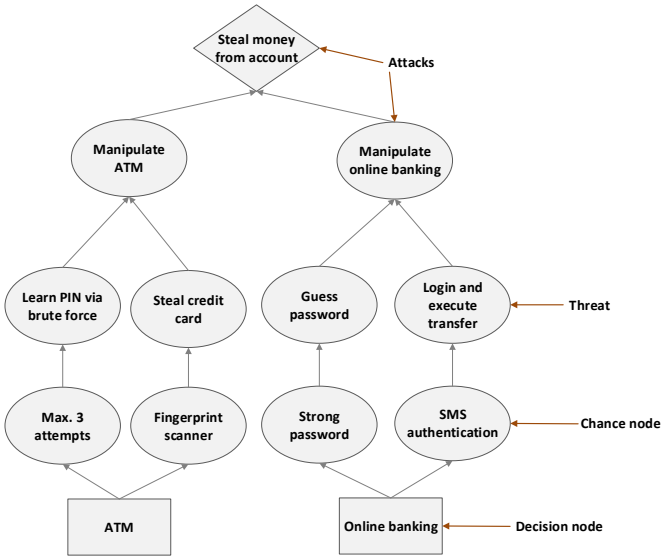


Figure 2.5.: Exemplary EID: *Steal money from account* [KW18] [ES09]

Let us recall the example of Figure 2.4 which we want to transfer into an semantically well-defined EID. In contrast to the ADT the EID is build bottom-up. First, there are two decision nodes *ATM* and *Online banking*. In case of *ATM* there are two chance nodes, i.e. CMs: *Max. 3 attempts* and *Fingerprint scanner*. By means of these chance nodes the threats *Learn PIN via brute force* or *Steal credit card* are mitigated. In this way the superordinated attack, i.e. chance node *Manipulate ATM* is mitigated. This chance node as well as the chance node *Manipulate online banking* leads to the root threat *Steal money from account* and is represented by means of a utility node. The scenario described above including accomplishment of attack *Manipulate online banking* is depicted in Figure 2.5. [KW18]

2.3. Modelling Notations

In computer science, it is often essential to express knowledge in modelling notation. This section brings the GSN as well as the FM more closer.

2.3.1. Goal Structuring Notation

Nowadays, in safety-critical environment it is task of developers to document for certification authority that a system accomplishes some degree of safety. The arguments, which are used to describe that, are called Safety Case (SC). Tim Kelly and Rob Weaver defined a SC as follows:

Definition 2.11 (Safety Case)**.** "A Safety Case should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context." [KW04]

Thereby, it is purpose of the SCs which are usually used from authorities to phrase it in a *clear* way. *Arguments* are used for describing how to get safety based on the safety evidences. The *system* to describe must acceptably be safe since 100 % of safety is practically impossible. Moreover, it is important that the SC is used in the right *context*. Otherwise the system may be unsafe. [KW04] Tim Kelly and Rob Weaver described in their paper [KW04] that it is not sufficient to list all requirements including their context and evidence, e.g. using FMEA or FTA. Therefore, it is essential to evince relationships between those elements and to provide convincing argumentation on how claims and requirements are supported by evidence and in which context they apply. Figure 2.6 visualises the situation in a graphical manner. It is therefore purpose of the GSN to communicate SCs in a standardised argumentation notation that can be used and understood from everyone to describe safety of safety-critical systems. [Spr12] In that way it is ensured that all stakeholders have the same understanding of the safety arguments presented in [KW04].

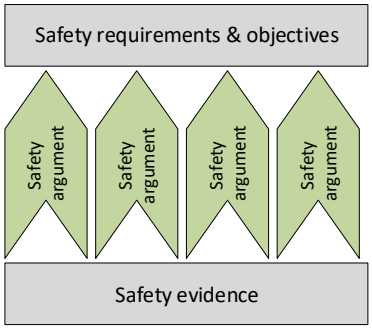


Figure 2.6.: Safety argumentation [KW04]

It is the aim of the GSN to describe SCs in a well-defined, standardised, precise and structured way. Therefore, Tim Kelly and Rob Weaver [KW04] defined several graphical object nodes as well as two kinds of associated relationships. A listing of all object types including their functionality is given in Table 2.9 and Table 2.10. Claims, requirements and arguments are represented by goals whereas evidence is realised in terms of solutions. By using strategies it is specified which approach has been taken to apply an argument. Moreover, it is possible to specify the context for which the claims are applicable. Furthermore, accepted assumptions are also considered by means of the GSN. Additionally, justifications are realised to take evidence or regulations into account. Finally, the GSN objects have to be interconnected to each other to get a well-defined goal structure.

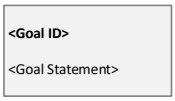
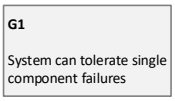
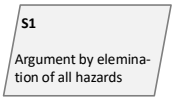
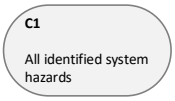
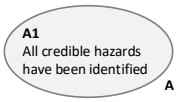
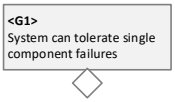
GSN element	Example	Semantics
		A **Goal** in GSN is a claim as a part of an argument.
		A **Strategy** in GSN describes how an argument is being presented.
		A **Solution** in GSN provides evidence to support a claim.
		A **Context** in GSN describes the circumstances for which the argumentation holds.
		An **Assumption** in GSN is a fact the argumentation relies on.
		A **Justification** in GSN explains why the argument should be accepted.
		Represents an **undeveloped** claim. Can be attached to goals or strategies to indicate that this claim hasn't been developed further intentionally.

Table 2.9.: Graphical objects in GSN [KW04]

GSN relation	Semantics
	The **in-context-of** relation in GSN references the context or requirements of goals or strategies. It allows the following relationships: • Goal to {context, assumption, justification} • Strategy to {context, assumption, justification}
	The **supported-by** relation in GSN defines inferential and evidential correlation. The following GSN elements can be connected: • Goal to {goal, strategy} • Strategy to goal

Table 2.10.: Relations in GSN [KW04]

Figure 2.7 and Figure 2.8 show an introducing example using Brake-by-Wire (BbW) system, inspired by the TIMMO-2-USE project [Per+12]. A BbW is performed by a electronically controlled brake-system of an automotive vehicle where the brakes are operated by sensors and actuators which are monitored by ECUs. BbW systems have no hydraulic or mechanic coupling between brake pedal and the actual brakes. Their connection and functionality rather depends on an electronic bus-system. For this purpose, the individual components of the brake are cross-linked by ECUs.

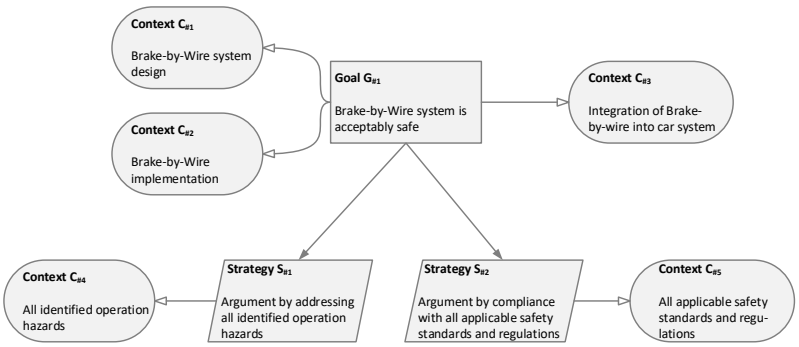


Figure 2.7.: GSN Example: BbW part 1

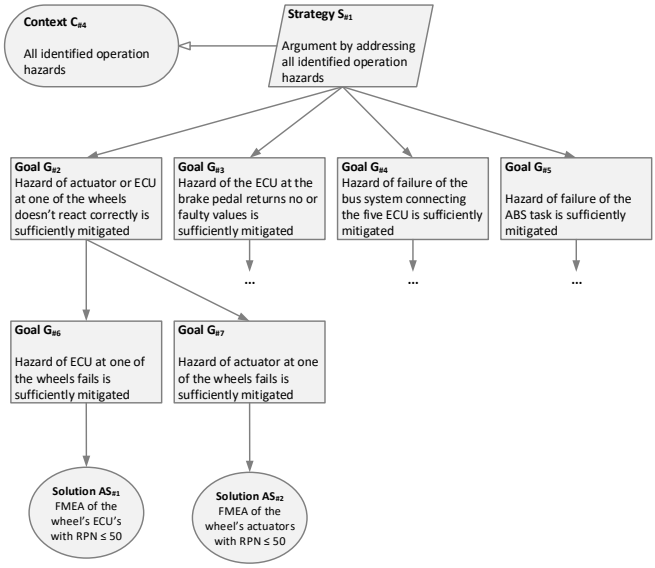


Figure 2.8.: GSN Example: BbW part 2

Goal $G_{\#1}$ in Figure 2.7 defines the root goal, i.e. main goal of the BbW SC. In this context, it claims that the *BbW system is acceptably safe* taking system design, implementation and system environment (cf. context $C_{\#1}$ to $C_{\#3}$) into account. Furthermore, there are two strategies (strategy $S_{\#1}$ and $S_{\#2}$) how to fulfil goal $G_{\#1}$ in this context. The first strategy is illustrated in Figure 2.8 and clarifies how to

identify operation hazards and how they are mitigated. For this purpose, goals $G_{\#2}$ to $G_{\#7}$ are responsible for this and address the actuator, ECU, bus system and Anti-lock Braking System (ABS). To prove that goals $G_{\#6}$ and $G_{\#7}$ are being achieved, the solutions $AS_{\#1}$ and $AS_{\#2}$ are applied which both depend on FMEA to consider risk assessment. The example showed that the GSN notation uses a simple and clear description which is easy to understand and realise.

2.3.2. Feature Modelling

Nowadays, systems are getting more complex and thus it is quite difficult to make them controllable. For this purpose, clearness regarding structure of a system has to be enabled. In this context, SPL orientated design and development has been established to allow variability. The necessity of SPLs is covered later on in Section 5.1. By means of proper structuring reusability and variation of sub-structures are possible. [Böc+04] Figure 2.9 illustrates an exemplary basic FM with all possible core elements, relationships and an extension in the context of this thesis. The core element of FMs are features. They constitute sub-components of the whole system and are structured hierarchically. Therefore, the FM is structured like a tree diagram and can be splitted in easy understandable and still detailed sub-systems. Hereinafter, the functionality of FMs is explained in more detail.

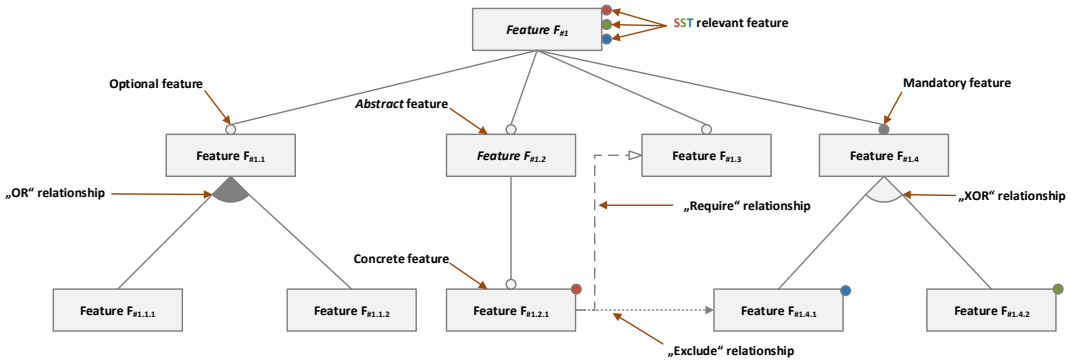


Figure 2.9.: Elements of a FM

In general, FMs consists of features, relationships and additional characteristics. The purpose of features has already been explained in previous paragraph. Furthermore, there are two types of relationships: Hierarchical relationships and cross relationships between features of different hierarchical layers. Within the hierarchical relationships there are two types: *OR* and *ALT* relationship. The first one indicates that at least one feature needs to be selected. The *ALT* relationship, which is also called *XOR* relationship, specifies that exactly one feature is part of the parent feature. The second kind of relationship is differentiated in two types: *requires* and *exclude* relationship. As the name suggests, these relationships require the existence of another feature or exclude other features. By using these relationships it is possible to connect features between several layers, i.e. the

features do not necessarily have to be connected hierarchically. The semantics of a parent-child relationship is interpreted as *feature a can be part of feature b* and not as *feature a is part of feature b*. In this way, reusability and variation is enabled. Furthermore, there are two characteristics that describe the dependency of a feature of its sub-features: *Optional* and *mandatory*. *Optional* is the default value and *mandatory* specifies that this feature must be selected. [CHE04] Moreover, the classic FM has been extended for safety-critical functionality. It is now possible, to annotate features with a safety, security or timing flag. In this context, structuring of safety-critical features can be performed more precisely. For this purpose, three different coloured dots mark the features (cf. note of *Feature 1* in Figure 2.9). The graphical notation of FMs are standardised uniformly but the SST flags. This notation can be taken from Figure 2.9.

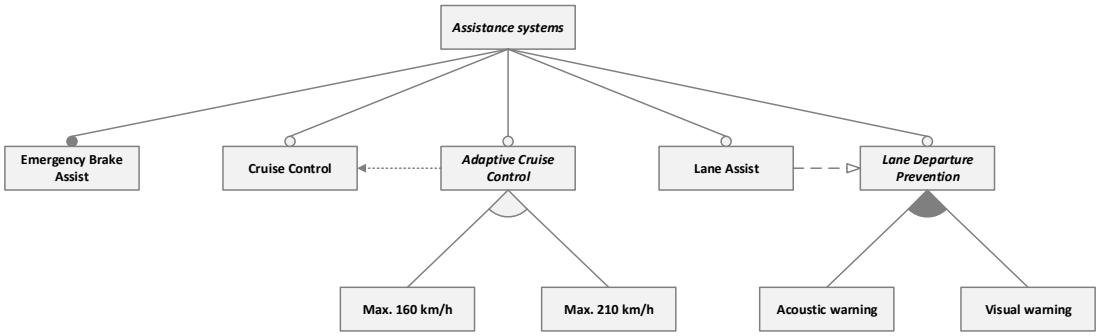


Figure 2.10.: Exemplary FM: ADASs

Figure 2.10 shows a real example of the automotive industry. In focus here is the configuration of ADASs. The customer can select between five different ADASs: *EBA*, *Cruise Control (CC)*, *ACC*, *LA* and *LDP*. It has to be noted that the EBA is mandatory whereas the other ADASs are optional features. The *ACC* excludes the CC feature since the ACC includes CC functionality. If the customer orders an *ACC* he or she must decide between a system that allows a maximum permissible speed of 160 km/h or 210 km/h. Furthermore, the potential customer is forced to choose LDP if LA has been selected. For the LDP system there are two available options of which at least one has to be enabled: Acoustic and visual warning. As one sees by means of FMs you can create product architectures that represent a product to be manufactured.

2.3.3. System Modelling

Nowadays, hardware and software systems are becoming increasingly complex and extensive by using a large number of components. Let us recall an example from the automotive industry. As mentioned in Section 1.1 modern cars are equipped with a variety of (semi-)autonomous ADASs. Therefore, many ECUs and sensors are installed. For this reason, it is essential to model involved system

components in a manageable manner. First, the term *system* is defined. Subsequently, it will be explained how system modelling is specified.

Definition 2.12 (System)**.** According to [Sch99] a *system* is characterised by availability of certain features and by the following four axioms:

1. **Structural principle:** The system consists of several parts which are related mutually by each other and the system environment.
2. **Decomposition principle:** The system consists of several parts which can be decomposed in related sub-parts. Each sub-system has different system characteristics.
3. **Principle of causality:** The system consists of several parts whose relationships and amendments between them are determined clearly.
4. **Temporal principle:** The system consists of several parts whose structure and state is determined by temporal procedures and amendments.

According to [Jan10] it is the main task of system modelling to organise all the different system components of complex systems. Furthermore, a system is classified as a complex system if there is a variety of system components. Moreover, human beings are just able to keep the overview of a limited number of elements, i.e. graphical symbols and their semantics. This is reflected by system modelling Therefore, there are four design principles to keep also the overview of very complex systems which has been presented by [Jan10]:

- **Structuring:** The system is decomposed as far as possible according to specific criteria. In this way, relationships are identifiable.
- **Decomposition:** The system is decomposed into basic components and subsystems. Hence, there are more details available in the current model and the structure is finely granulated.
- **Aggregation:** Sub-systems are aggregated into an entire system by merging single elements. Aggregation is consequently the opposite of decomposition.
- **Hierarchy:** In general, system definitions are hierarchical, i.e. sub-systems can be considered as new systems. The first hierarchical level presents a global view on the system whereas the bottom layer enables a detailed view on the system.

Figure 2.11 represents hierarchical levels of a SM. There is a decomposition from layer 0 $\rightarrow$ layer 1 and an aggregation from layer 1 $\rightarrow$ layer 0 [Jan10]. There are several graphical modelling notations for system modelling The corresponding notation used in this thesis is presented in Section 3.3.2.

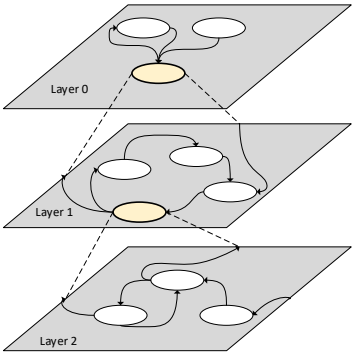


Figure 2.11.: Hierarchical layers of a SM [Jan10]

2.4. Multi-Criteria Decision Making

Frequently, we were facing the problem that we have to take a decision between more given alternatives. However, the decisions are not always easy to make. This is due to the fact that we often prefer attribute 1 from alternative 1, attribute 2 from alternative 2 and so on. Therefore, an intelligent algorithm is needed which allows decision making by taking individual preferences into account. In this chapter three common techniques are presented: The AHP, Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) and Utility Analysis (UA).

For a better understanding of the three MCDM algorithms a simple example is introduced. Let us assume we need to decide between three cars *Car a*, *Car b* or *Car c*. There are the attributes *power*, *number of installed safety features* (in the course of Section 2.4.1 to Section 2.4.3 abbreviated as: *safety*) and *price* that have to be assessed. The *Car a* has the best price whereas the *Car c* has the highest number of safety features and best power. In turn, the *Car b* has the best average values.

The following three sections give an answer how to calculate the optimal solution by applying the AHP, TOPSIS or UA.

	Power	**Safety**	**Price**
Car *a*	190 HP	19	37.250 €
Car *b*	231 HP	43	62.850 €
Car *c*	286 HP	74	90.600 €

Table 2.11.: MCDM example: Alternatives and attributes

2.4.1. Analytic Hierarchy Process

As indicated in the preceding section, decisions often have to be made on the basis of several criteria. However, these criteria depend on further other criteria. Therefore, the AHP is used for complex and nested criteria and structures the

attributes in a hierarchical way, i.e. there is a top-level attribute with a lot of sub-attributes in different levels. For the case study we need just one layer, because there are just the attributes *power*, *safety* and *price* that have to be compared. For that purpose the corresponding attributes are compared pairwise. Thereby, values between *1* and *9* are assigned. *1* means that attribute x is just as important as attribute y, whereas *9* means that attribute x is much more important than attribute y. [Tri00] A detailed description is gathered in Table 2.12. If attribute x compared to attribute y is assigned value z, it is obvious that attribute y compared to attribute x is assigned the reciprocal value $\frac{1}{z}$. [Saa04] The pairwise comparison in general as well as of the running example can be taken from Table 2.13. The notation of using matrices with the extension of rows and columns has been taken from [Saa04]. In this example the following applies: *Safety* (A_2) $\succ$ *price* (A_3) $\succ$ *power* (A_1).

Intensity of importance	**Definition**	**Description**
1	Equally important/liked	Both criteria contribute equally to the objective
3	Moderately more important/preferred	One criterion is favoured over the other
5	Strongly more important/preferred	One criterion is largely favoured over the other
7	Very strongly more important/preferred	One criterion has been proved to dominate the other
9	Extremely more important/preferred	One criterion is favoured over another based on evidence of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values	Compromise between the two adjacent judgements

Table 2.12.: AHP: Scale of relative importance [SK90]

First, we need a square matrix to perform a pairwise comparison. The matrix is square since the attributes $A_1, A_2, ..., A_n$ are compared row by column respectively. When making judgements in form of pairwise ratios we have to ensure that we do not use inconsistent ratios, i.e. transitivity must be ensured. [Saa04] Concretely, the following applies:

Definition 2.13 (Transitivity)**.** There is a relation $R_\succ \subseteq A \times A$ with set A of attributes . Then, the following applies for observing transitivity [GV17]:

$$\forall x, y, z \in A : xR_\succ y \ \wedge \ yR_\succ z \ \Rightarrow_T \ xR_\succ z$$

According to [Saa04] it is necessary to check that the matrix is not inconsistent, i.e. the consistency ratio CR has to be calculated and may not exceed 10 %. For that

purpose, the consistency index CI is determined that depends on the maximum eigenvalue λ_{max} and the dimension n of the matrix. Furthermore, CR depends on an average consistency index CI_R of 500 random generated matrices that has been performed by [Saa04]. In summary, the consistency ratio can be calculated as follows:

Definition 2.14 (AHP Consistency Ratio)**.** The consistency ratio CR of a matrix A with the dimension n is defined as follows [Saa04]:

$$CR = \frac{CI}{CI_R} = \frac{\frac{\lambda_{max}-n}{n-1}}{CI_R} \leq 0,1$$

$$A = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \dots & A_n \end{array} \\ \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \dots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \dots & \frac{w_2}{w_n} \\ \dots & \dots & \ddots & \dots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \dots & \frac{w_n}{w_n} \end{bmatrix} \end{array} \quad A = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{bmatrix} 1 & \frac{1}{6} & \frac{1}{3} \\ 6 & 1 & 4 \\ 3 & \frac{1}{4} & 1 \end{bmatrix} \end{array}$$

Table 2.13.: AHP matrices with weights [Saa04]

According to Definition 2.14 the consistency ratio of the example is:

$$CR = \frac{\frac{3,05-3}{3-1}}{0,58} \approx 0,043$$

Consequently, it has been shown that the matrix is consistent. Subsequently, it is continued with calculating of the weights w_i. It first requires that the matrix of Table 2.13 is normalised. Based on the normalised matrix the weight w_i of attribute A_i can be calculated:

Definition 2.15 (AHP Weight)**.** The weight w_i of attribute i within a matrix A with dimension n and a row total r_i is defined as follows [Saa04]:

$$w_i = \frac{r_i}{n}$$

By applying Definition 2.15 to the example there are the local priorities that are listed in Table 2.14. The results reflect our prerequisite that *safety* $\succ$ *price* $\succ$ *power*. To determine our best alternative for our example with respect to the information already made it is still required to calculate the local priorities of the individual attributes for all alternatives.

	Power	**Safety**	**Price**	r_i	w_i
Power	0,1	0,12	0,06	0,28	0,09
Safety	0,6	0,7	0,75	2,05	0,68
Price	0,3	0,18	0,19	0,67	0,22
Σ	1	1	1	3	1

Table 2.14.: AHP example: Normalised matrix and weights w_i

To calculate the local priority v_{ij} of alternative i and attribute j the percentage distribution is determined depending of the real measurable values, e.g. in the running example the HP [Saa04]. The formal definition of the local priority therefore is:

Definition 2.16 (AHP Local Priority)**.** The local priority v_{ij} of alternative i and attribute j with real measurable value u_{ij} is defined as follows [Saa04]:

$$v_{ij} = \frac{u_{ij}}{\sum_{j=1}^{n} u_{ij}}$$

By applying Definition 2.16 to determine local priority of alternative *Car a* and attribute *power* one gets:

$$v = \frac{190}{190+231+286} \approx 0,269$$

The complete list of all local priorities can be taken from Table 2.15. Finally, the last step of the analysis is performed, in particular calculating the global priorities of the individual alternatives i. For this purpose, the already calculated weights according to Definition 2.15 or Table 2.14 as well as the local priorities according to Definition 2.16 or Table 2.15 are used (see Definition 2.17).

	Power	**Safety**	**Price**	**Global priority**
Car *a*	0,269	0,140	0,403	0,208
Car *b*	0,327	0,316	0,335	0,318
Car *c*	0,405	0,544	0,262	**0,464**
Σ	1	1	1	1

Table 2.15.: AHP example: Local and global priorities

Definition 2.17 (AHP Global Priority)**.** The global priority π_i of alternative i and attribute j with weight w_j and local priority v_{ij} is defined as follows [Saa04]:

$$\pi_i = \sum_{j=1}^{n} w_j \cdot v_{ij}$$

The global priorities of the running example can be also taken from Table 2.15. By means of the attached table it can be concluded: *Car c* $\succ$ *Car b* $\succ$ *Car a*. This can be explained by the fact that the number of installed safety features has been weighted the highest in this example. The alternative *Car b* is the second best option since it has the best average values for all three attributes. As it can be seen the AHP algorithm is based on likelihood calculations and easy to understand. One major advantage is that the algorithm is built up hierarchically, i.e. it is possible to nest the attributes to refine them. However, there is a disadvantage: If there are matrices with more than five attributes it is quite difficult to keep consistency ratio under control since the matrices are created manually by experts in most cases.

2.4.2. TOPSIS

The TOPSIS has been developed by Hwang and Yoon in 1981 with the objective of an easy understandable technique for efficiency analysis. Furthermore, TOPSIS is applied in various domains, e.g. assessment of transport systems, selection of robots for industrial use or operational location selection. [PZ07] First, the TOPSIS defines the criteria that has to be taken into account. In this context, it is, similar to the AHP differentiated between cost criteria and benefit criteria. For cost criteria the lowest value is the best one whereas for benefit criteria it applies vice versa. TOPSIS does not define how to assess relative criteria importance. Therefore, established methods, e.g. the AHP has to be applied. [PZ07] Let us assume there is an assessment of criteria importance performed with the AHP that is listed in Table 2.14. On the basis of these data it should be determined by means of TOPSIS which alternative of the example is the optimal solution. The first step is calculating local priorities of the individual alternatives. This is done by applying Definition 2.16. The results are identical to the AHP and are listed in Table 2.15.

Definition 2.18 (TOPSIS Best and Worst Case Alternative)**.** According to [PZ07] the best case alternative A^+ and worst case alternative A^- of alternative i and attribute j is defined as follows :

$$A^+ = \{(\max_i = \{v_{ij} \mid j \in J\}), (\min_i = \{v_{ij} \mid j \in J'\}) \mid i = 1, ..., n\} = \{v_1^+, v_2^+, ..., v_j^+, ..., v_m^+\}$$

$$A^- = \{(\min_i = \{v_{ij} \mid j \in J\}), (\max_i = \{v_{ij} \mid j \in J'\}) \mid i = 1, ..., n\} = \{v_1^-, v_2^-, ..., v_j^-, ..., v_m^-\}$$

where

$J = \{j = 1, ..., m \mid \text{criterion } j \text{ belongs to benefit criteria}\}$

$J' = \{j = 1, ..., m \mid \text{criterion } j \text{ belongs to cost criteria}\}$

and

$J \cap J' = \emptyset \ \wedge \ J \cup J' = \{1, ..., m\}.$

Subsequently, it is necessary to determine the best case and worst case alternative A^+ and A^- of each attribute. The calculation is laid down in Definition 2.18. Hereinafter, the values are summarised in Table 2.16.

	Power	**Safety**	**Price**
A^+	0,405	0,544	0,403
A^-	0,269	0,140	0,262

Table 2.16.: TOPSIS example: Best/Worst case alternative of each attribute

Based on the ascertained best case and worst case values the clearances S_{i^+} or S_{i^-} depending on A^+ or A^- must be defined for each alternative A_i. The clearances are so-called Euclidean distances. [PZ07] The subsequent definition summarises calculation of S_{i^+} and S_{i^-}.

Definition 2.19 (TOPSIS Clearances)**.** The clearances S_{i^+} and S_{i^-} of alternative i is defined as follows [PZ07]:

$$S_{i^+} = \sqrt{\sum_{j=1}^{m} (v_{ij} - v_j^+)^2} \qquad \forall i = 1, ..., n$$

$$S_{i^-} = \sqrt{\sum_{j=1}^{m} (v_{ij} - v_j^-)^2} \qquad \forall i = 1, ..., n$$

where

v_{ij} refers to the local priority of alternative i and attribute j and

$v_j^{+/-}$ refers to the corresponding BC or WC value of A^+ or A^-).

By applying Definition 2.19 to the example the clearances of Table 2.17 are given. This is not the final result which is analysable. For this purpose, we need the distance index C_i^+ for relative proximity to the best case alternative A^+. C_i^+ only depends on S_i^+ and S_i^- and is defined in Definition 2.20 [PZ07]. Finally, the optimal alternative can be determined by applying them. Analogous to the AHP algorithm the result is the same: *Car c* $\succ$ *Car b* $\succ$ *Car a*. This can be justified by the fact that the alternative *Car c* has the best case attribute twice. Alternative *Car a* has one best case attribute, but it has two worst case attributes. Therefore, *Car b* is the second best choice.

	S_i^+	S_i^-	C_i^+
Car *a*	0,426	0,141	0,249
Car *b*	0,250	0,200	0,440
Car *c*	0,141	0,426	**0,751**

Table 2.17.: TOPSIS example: Results

In summary, the TOPSIS is a suitable MCDM algorithm and requires little prior knowledge. However, it requires priorities of assessment criteria are already be present. [PZ07] recommend to use AHP for that. Therefore, it is more practicable to apply AHP instead.

Definition 2.20 (TOPSIS Best Case Distance Index)**.** The distance index C_i^+ of alternative i for relative proximity to best case alternative A^+ is defined as follows [PZ07]:

$$C_i^+ = \frac{S_i^-}{S_i^+ + S_i^-} \qquad \text{where } 0 \leq C_i^+ \leq 1 \qquad \forall i = 1, ..., n$$

2.4.3. Utility Analysis

The UA is a scoring model that helps to make decisions more easier, especially in macroeconomics, project management or controlling. However, it is applicable in various domains including embedded and SCSs. Analogous to the AHP and TOPSIS it is essential for the UA to determine which attributes are most important. In contrast to the two MCDM methods that have already been presented, the UA is kept simple to save time and to perform it in an easy manner. For that purpose, we create a table consisting of the individual alternatives, criteria and (weighted) scorings. In this context we want to find out which type of cars is the optimal decision regarding the preferences that have already been listed in Table 2.14. The scoring v_{ij} of alternative i and attribute j in the example consists of values between the interval $1 \leq v_{ij} \leq 5$. [EWL10] Conversely, this means that an alternative i has a score of 5 points in best case and 1 in worst case. The best scoring of alternative *Car a* is *price* (4) whereas *safety* is the worst attribute (2). Alternative *Car b* has average to above-average scorings (3-4). The third alternative *Car c* scores with the attributes *power* and *safety* (5 respectively) whereas price is below the average (2). The complete list of all (weighted) scorings is listed in Table 2.18. The calculation of the overall result of the alternatives can be performed by applying Definition 2.21 and has been listed in Table 2.18.

		Car *a*		**Car *b***		**Car *c***	
Criterion	Weight	Score	Weighted	Score	Weighted	Score	Weighted
power	0,09	3	0,27	4	0,36	5	0,45
safety	0,68	2	1,36	3	2,04	5	3,40
price	0,22	4	0,88	3	0,66	2	0,44
Σ	1		2,51		3,06		**4,29**

Table 2.18.: UA example: Results

As it can be taken from Table 2.18 the UA concludes that *Car c* $\succ$ *Car b* $\succ$ *Car a*. This can be justified by the fact that alternative *Car c* has the best scorings in *safety* and *price* whereas *price* has highest priority. Alternative *Car b* has (above) average scorings for all attributes, therefore it is the second best alternative.

Definition 2.21 (UA Overall Scorings)**.** The overall scoring σ_i of alternative i and attribute j is defined as follows [EWL10]:

$$\sigma_i = \sum_{j=1}^{n} w_{ij} \cdot v_{ij}$$

where

w_{ij} represents the weighting and

v_{ij} reflects the corresponding scoring.

Car a only has a good *price*, *power* and *safety* is better for the competitive alternatives. In summary, the UA can be applied with little prior knowledge and for less extensive decision makings. Furthermore, the UA should be used for decisions with a small level of detail. Therefore, it is recommended to use AHP or TOPSIS for complex MCDMs since the level of detail is more exactly.

Part II.

Multi-Concerns Engineering for Safety-Critical Systems

3

Multi-Concerns and Multi-Criteria Decision Making

The current chapter describes one of the three main parts, the MCDM in more detail. First, the term *trade-off* serves as a basis and is specified in Section 3.1. Subsequently, the underlying concept of this chapter including a concept picture is explained in Section 3.2. Section 3.3 specifies necessary prerequisites of the MCDM. Modelling of goal hierarchy is specified in Section 3.4 and covers creating a SSTM and the necessary modelling for further security analysis which is also part of the MCDM. Section 3.5 gives a detailed description how to define relative importance of the SSTM components. In this context, the consistency ratio, i.e. transitivity property plays an important role and needs to be improved in some cases. It is part of Section 3.6 to describe how the consistency ratio can be improved. The essential risk assessment of the MCDM is described in Section 3.7. In this context, it is differentiated between safety and security risk assessment. The proper analysis algorithm is presented in Section 3.8. There are two algorithms available, the PCM and RCM. For a better understanding the MCDM is explained by means of a selected example which is described in Section 3.9. Finally, an overview of related work is given. The content of this chapter is mainly based on [LFB18] and [Fen16] which has been written by the author of this thesis and fellow researchers and is not cited any more

3.1. Trade-Offs

The development and modelling of safety-critical embedded systems demands some considerations, dependencies and objectives including safety, security and real-time requirements. As already indicated in Section 2.1.4, these demands are usually of different importance and are partially conflicting. Consequently, aspects which have been realised in consideration of safety or security issues possibly cause another safety-critical risks. Thus, solving a safety or security vulnerability might induce severe safety problems. Nowadays modern cars are equipped with a large number of wired and wireless interfaces. According to the trend in the automotive domain wireless communication increases immensely by supporting car2X communication. By transmitting most current and very important information, e.g. warnings, weather conditions or traffic news to vehicles driving within close proximity road safety is influenced significantly. Wireless

communication offers high risk of hacking attacks. Therefore, the functionality of modern automotive vehicles are highly safety-critical. Even minor interferences can impact functionalities tragically. Although car2x communications provide safety improvements nowadays, it still offers hacking attacks regarding safety and security issues. These potential problems seem to be solved by applying a secure data encryption. However, such an encryption causes at this point new timing problems, since more time is needed to execute the tasks. Let us remind the example which has been introduced in Section 1.1. The Ford Motor Company decided in the late 1960ies to develop a car which is cost-effective as well as fuel-efficient. Thereby, safety aspects and goals have been neglected due to lack of time. This decision has proved as wrong decision afterwards since the consequential costs (procedural costs, compensations costs and so on) were much higher than the costs that would have been incurred for safety-critical development of the car. [Ord+09] In principle, a car manufacturer operates on the principle of profit maximisation. To develop a fuel-efficient, cost-efficient as well as a safe car, it is necessary to evaluate these three criteria each other for determining an optimal result, i.e. maximisation of all three criteria. In this particular case the Ford Motor Company would have needed an optimal trade-off between safety, fuel efficiency and cost efficiency whereas safety should have highest priority. As can be seen, it is not trivial to find an appropriate solution to fulfil SST requirements to the highest degree. For this purpose, a suitable trade-off has to be found which covers all the functional and qualitative requirements to the best possible level.

Definition 3.1 (Trade-Off)**.** A trade-off reflects an optimal decision based on criteria. If each criterion would be assigned the best possible value, it might be feasible that the criteria are excluding each other.

In this thesis, trade-offs are the result of MCDMs and require modelling and specifying of safety-critical aspects and concerns including their dependencies beforehand (cf. subsequent section). SCSs are used in different application domains or industries such as automotive, avionics or railway. Nowadays, the trend is shifting towards autonomous driving. To enable autonomous driving a variety of ADASs and thus some hardware components are required and must be installed. Usually, the individual components have different SST requirements. The reason is that each hardware component has its own purpose. For instance, the realisation of an ACC may require a radar sensor and a camera-based sensor. The requirement of a radar sensor would be that the range is sufficient to determine the distance to the vehicle driving ahead. One of the main challenges of camera-based sensors is the recognition of objects despite back-lighting of the sun. In this case, a trade-off has to be determined corning safety requirements. In practice, security and timing issues must be considered as well and complicate decision making process. Hereinafter, some examples of SST requirements are introduced which need to be fulfilled by safety-critical components:

- *Safety:* Range of sensor is sufficient, object recognition works fine despite back light of the sun.
- *Security:* Data traffic is protected against third parties, ECUs are protected against external manipulation.
- *Timing:* Brake command is performed in time, airbag is triggered in time.

3.2. Concept

This section presents the concept of the approach, i.e. the MCDM in detail. Thereby, a concept picture, which is illustrated in Figure 3.1, is used initially to describe the overall approach of the MCDM in an abstract manner. Furthermore, the necessary steps, which are used in the concept, are explained in more detail.

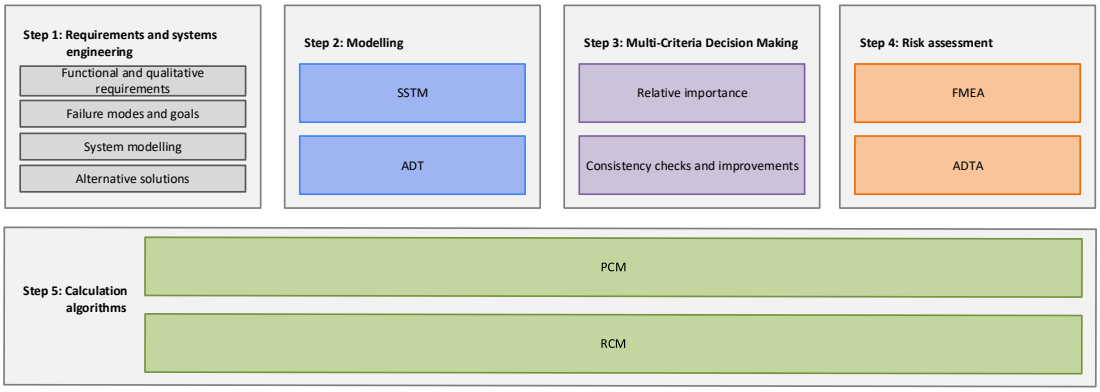


Figure 3.1.: Logical concept of the MCDM

When developing SCSs the design phase of the MC engineering process is necessary with respect to contradicting SST issues which have already been explained in Section 3.1. A MCDM is developed to avoid SST conflicts:

1. Devise potential alternative system designs
2. Identify and structure SST objectives
3. Perform risk analysis
4. Apply MCDM to find the safest solution

In general, all steps of the concept picture are covered within the current chapter. The individual steps or order can be taken from the boxes of the concept picture and are described hereinafter. A detailed explanation of the individual steps is done in further course of this chapter.

Step 1. As usual in each software development process it is first essential to define requirements and to apply systems engineering. Otherwise it is not possible to calculate an optimal trade-off. For this purpose, it is necessary to define proper SST requirements. The requirements are available in GSN notation and define occurring failure modes of the SCSs taken SST aspects into account. Failure modes and goals have to be determined for all MC, i.e. SST issues. Furthermore, it is essential to collect all system specifications within a SM which contains primarily software and hardware components and dependencies between them. Moreover, it is mandatory to specify alternative solutions for the MCDM, i.e. possible decision options between which the MCDM process have to be ruled. Defining failure modes, goals and SM depends on requirements, whereas specifying alternative solutions result from the underlying SM.

Step 2. In general, this step covers modelling and processes collected data of step 1 to perform the approach of the MCDM. This step includes modelling of SSTM and ADT. The SSTM contains the individual failure modes, goals and alternative solutions in an hierarchical structure whereas the ADT concerns modelling of security attacks and relating thereto suitable CMs.

Step 3. In this step the AHP algorithm is performed including determining the relative importance of the goals and POVs within the SSTM. In this way, the SSTM is used as input for the AHP and is refined by step 4 and 5 to get the output. For this purpose, it is necessary to compare the individual goals or POVs of the SSTM by each other to determine local and global priorities. In this context it is also necessary to check consistency ratio and, if applicable, to update relative importance to get an acceptable consistency.

Step 4. The second step deals with performing risk assessment using FMEA and ADTA technique. These methods cover SST issues whereas the ADTA is only responsible for security concern. Economic aspects are also covered by the ADT based ADTA. By means of the FMEA technique probability of occurrence, severity and detection are taken into account for the interrelations between individual POVs and alternative solutions.

Step 5. This step describes the calculation of the optimal trade-off based on information of step 1-4. In this context, two algorithms are available, the PCM and the RCM. The first one is mainly based on AHP calculations whereas the second one is mainly based on risk assessment calculations.

3.3. Requirements and Systems Engineering

As usual for every software development process, it is necessary before the design phase to perform requirements engineering. Besides requirements engineering there is a change request process. The requirements engineering process is covered

in this chapter whereas the change request process is part of Chapter 4. The requirements engineering process is defined in the following order:

1. The necessary requirements are identified.
2. The requirements are analysed and reviewed and if necessary identified once again.
3. The requirements will be specified until they are validated. However, if the validating process fails a repeated specifying process is necessary.

This procedure has to be repeated for all requirements. [Som11] A detailed description of this process is given in Section 3.3.1. In general, there are two types of requirements namely functional requirements and quality requirements:

- *Functional requirements* define functionality and variability of a software system. Typically, these requirements are defined from function view (input/output of a system, failure situations), data view (data structures, integrity conditions) and behaviour view. [Poh10]
- *Qualitative requirements* on the other hand define criteria for goodness of a software system or individual system components. These include requirements, e.g. SST requirements, reliability, usability, performance, changeability, portability or scalability. [Som11]

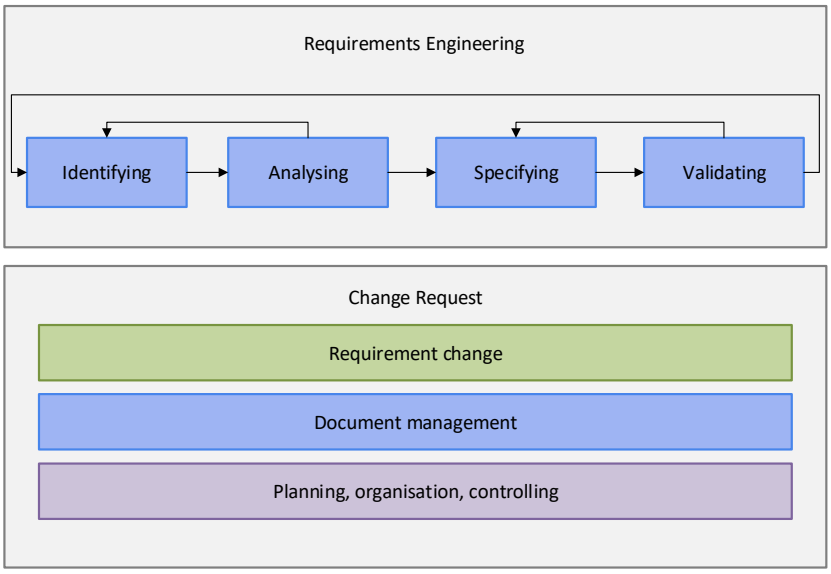


Figure 3.2.: Requirements engineering and management process [Som11]

The requirements engineering process is mandatory and prerequisite for the change request process. It is the purpose of the change request process to track amendments of requirements and to manage requirement documents during the

whole life cycle. Moreover, core activities of the requirements have to be organised and controlled. [Som11] A graphical overview of the requirements engineering and requirements management process is illustrated in Figure 3.2.

3.3.1. Requirements Definition

As already mentioned in the previous section, it is mandatory to define requirements correctly. It is the purpose of the MCDM to achieve a maximum degree of safety for a SCS. Therefore, we consider SST requirements:

- *Safety:* It must be taken into account that the target system must prevent hazards and may not endanger human life at any time (cf. Definition 2.2).
- *Security:* It is covered that the final system has to be protected against data manipulation, data theft and unauthorised access (cf. Definition 2.4).
- *Timing:* It is considered that timing intervals and thresholds are reached (cf. Definition 2.5).

Thereby, for each requirement certain conditions need to be fulfilled:

- *Technological:* The system and the technical IT infrastructure within which the software system should be executed and developed.
- *Organisational:* Structure and procedure organisation of stakeholders which use or develop the software.
- *Legal:* Compliance with laws, standards and norms.
- *Ethical:* Morals and wordings of the individual cultural circle, e.g. forms of address in which the system is applied. [Poh10]

In the previous section a rough roadmap for defining requirements has been presented. This process is explained in more detail by the following:

1. *Identifying:* It is essential to find existing as well as potential requirements to the SCS. Existing requirements can be determined by surveys or analyses whereas potential requirements are identified by creativity techniques, e.g. brainstorming or mind mapping. [Poh10]
2. *Analysing:* After identifying requirements they are usually unstructured. Therefore, it is necessary to classify and prioritise them. The classification of requirements entails clusters of coherent requirements. Furthermore, redundant and conflicting requirements are emphasised. Subsequently, the relevant requirements are prioritised upon consultation with the different stakeholders. [Poh10]
3. *Specifying:* This step transforms the analysed requirements in a default notation. For instance, requirements can be transformed into a graphical model notation. [Poh10]

4. *Validating:* This process is responsible for synchronising specified requirements with the identified requirements. In this way it is ensured that on basis of the resulting requirements the correct SCSs are developed by means of the MCDM. [Som11]

For the MCDM requirements are specified in textual form since they are transferred into a hierarchical goal structure by means of the requirements engineering process. The requirements are formulated by the different participated stakeholders. These include persons or organisations which are interested in the software system, e.g. software- and hardware specialists or regulatory authorities. [Poh10] The requirements for the MCDM have to be formalised with a high quality standard and consider all MC in order to get a significant result by the MCDM. For this purpose, it is mandatory to take SST into account by means of requirements. In context of the MCDM a security analysis is performed which analyses vulnerabilities of the SCS and analyses whether appropriate CMs are profitable or not. I.e., in this context the requirements have to be formalised in a way that quality attributes are fulfilled. Hereinafter, there are some examples of resulting SST requirements including their concerned stakeholders and the corresponding quality attributes in context of developing a safety-critical ACC system:

Stakeholders	**Requirements**	**Quality attribute**
Safety specialist, software developer	Distance to the obstacle is acceptably accurate	Safety
Data protection authority, software developer	Communication is acceptably secured against data theft	Security
Cyber security specialist, software developer	Messages arrive in time with acceptable reliability	Timing

Table 3.1.: Exemplary SST requirements for developing an ACC system

3.3.2. System Model

Section 2.3.3 described the necessary foundations of system modelling In this section, the SM, which is part of the systems engineering process, is used for the MCDM and is explained in more detail. For this purpose, a component diagram, which is used for system modelling of the MCDM, provides necessary information. In this context, the SM is used to derive necessary information for the MCDM. These information derive alternative solutions which are part of the following section.

Component diagrams are specified in the Unified Modeling Language (UML) which help to understand the structure of existing systems or to build new ones. *Components* are the major elements of the component diagram and are related with other system components of the component diagram. Thus, structural principle (cf. Definition 2.12) is fulfilled. Furthermore, components can be nested arbitrary

to specify the system on different abstraction levels as illustrated in Figure 2.11 of Section 2.3.3. Therefore, the component diagram also meets decomposition principle. Components are characterised by *interfaces* and *ports*. An interface is a collection of methods and attributes which describes the behaviour of a system component. There are two types of interfaces: *Required* and *provided interfaces*. A required interface requires an external interface from another interface, whereas a provided interface can be implemented by other interfaces for their use. A *port* usually clusters interfaces and defines an interaction point between the component and its environment. Since the port and interface connections are subject of causal relationship, the principle of causality is fulfilled. Moreover, creating a component diagram is governed by some temporal processes, e.g. specifying functional requirements. Therefore, the temporal principle as defined in Definition 2.12 is covered. [RQS12]

For the system modelling of the MCDM and for further steps of this thesis, the component diagram is reduced to merely provide relevant information for the analyses. For this purpose, we only model components and ports and link them with each other. This is due, among other reasons, to the fact that only components and ports are pertinent for the approach of this thesis since we need no implementation details, e.g. required or provided interfaces. In context of this thesis, the individual system components as well as how they are connected between them. Later in the course of this thesis, if we speak about SM it refers to the reduced SM which is described in this paragraph.

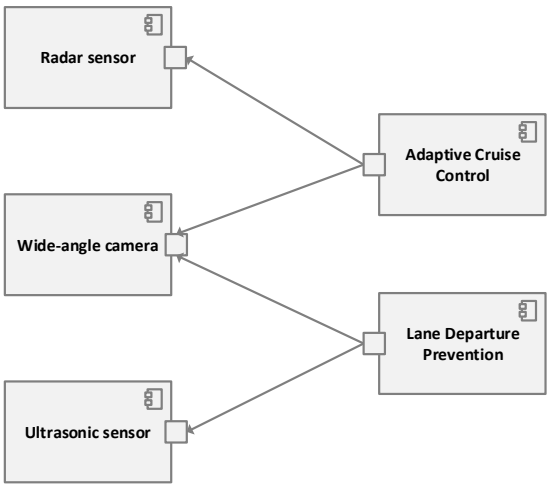


Figure 3.3.: Exemplary SM: Excerpt from ADASs

Figure 3.3 shows an exemplary and rudimentary excerpt from a SM which models relationships between two ADASs. There are two components representing the ADASs, namely *Adaptive Cruise Control* and *Lane Departure Prevention*. Furthermore there are three system components which represents the individual sensor types. These include *Radar sensor*, *Wide-angle Camera* and *Ultrasonic sensor*. By

means of the individual port connections, which are represented by small squares dependencies, can be determined. In this way, the system component *Wide-angle camera* is used from both, the *Adaptive Cruise Control* and *Lane Departure Prevention*.

3.3.3. Failure Modes, Goals and Solutions

In general, it is the purpose in each safety-critical domain, e.g. in the automotive, avionics or railway environment to develop a system which is as safe as possible. To enable this, some compromises, i.e. trade-offs have to be made since the individual aspects don't correspond with each other. For instance, if an airbag doesn't trigger in time it may endanger human life. To guarantee a maximum degree of safety the requirements engineering process (cf. Figure 3.1) is first mandatory to solve the problem. After specifying the requirements and modelling of system relationships by means of a SM it is mandatory to derive alternative solutions, failure modes and goals therefrom. These elements are described in more detail in the following paragraphs.

Failure Modes

The fundamental functionality of the MCDM is to define an hierarchical goal structure and to refine them step by step to consider SST aspects and how they are connected among each other. In this way, it can be ensured that the corresponding system is acceptably safe. It is the first step to define failure modes before specifying the corresponding goals. Without specifying any failure modes, i.e. identifying the manner in which the failures of a SCS occur, goals could not be defined correctly since necessary context information would be missing. Under certain circumstances, these context information could not be taken into account otherwise.

Goals

As already mentioned in Section 1.1, if it is neglected to define goals it may lead to economic, financial or personal damage. The Ford Pinto example showed that it can lead to serious consequences if there are no sufficient

a) defined goals

b) preventative risk assessments

The more precise the goals are defined the lower the probability of any damages or failures which can occur at a later time. Goals must be defined via different abstraction layers. The lower the degree of abstraction the more precise are the goals. [DLM13] There is one goal starting from root node, i.e. abstraction layer 0 which needs to be a safety goal since safety has highest priority. Proceeding from this goal, goals can be refined by further goals. Goals with an abstraction layer grater than 0 consider different concerns such as safety, security or timing,

i.e. security and timing influence safety in an indirect manner as mentioned in Section 2.1.4. Furthermore, non safety-critical aspects, e.g. economic aspects can be considered as well. These include, e.g. costs or profitability. However, in this thesis we only consider the three concerns SST unless they will mitigate safety risk. The first aspect is an essential part of the MCDM and has to be taken into account in the next paragraph. By means of preventative risk assessment it has to be clarified beforehand which kind of risks can occur and to which degree it may lead. For this purpose, the POVs need to be estimated with respect to the individual decision points of the MCDM, i.e. the alternative solutions (cf. next paragraph). Thereby, the probability of occurrence, severity and detection are of high importance. If the risk is not acceptable, the corresponding goals, POVs or alternative solutions have to be replaced or adapted.

Alternative Solutions

In automotive vehicles, air-planes or railways there are a number of installed hardware and software configurations whose interactions behave differently. Therefore, it is essential to apply the MCDM for each configuration set individually to get the best result. We call such configuration set *(alternative) solution*. Reversely, this means that the MCDM should analyse by means of a percentage distribution which solution is the most optimal taken SST requirements into account. To get a significant result there must be at least two solutions. Table 3.2 shows a set of possible solutions:

Solution	**Sensor(s)**	**Encryption**	**Bus system**
Solution $AS_{\#1}$	radar	64 bit	CAN
Solution $AS_{\#2}$	camera	128 bit	LIN
Solution $AS_{\#3}$	radar & camera	128 bit	FlexRay

Table 3.2.: Exemplary solution set for MCDM

In case of *solution* $AS_{\#3}$ the individual solution is interpreted as follows: The corresponding automotive vehicle (or something else) is equipped with a radar and camera based sensor. The corresponding data is en-/decrypted by means of a 128 bit encryption algorithm. Moreover, the data is sent via a FlexRay bus. Combining these three properties is the result of solution $AS_{\#3}$. Let us assume the MCDM is performed with the solution set of Table 3.2 the final result is a percentage distribution of solution $AS_{\#1}$, $AS_{\#2}$ and $AS_{\#3}$, i.e. summing up all solutions result 100%. The solution with the highest percentage distribution is that solution which best accomplish SST requirements. When selecting the corresponding solution system's safety properties are fulfilled optimally, i.e. if another solution with worse percentage value is applied the system is no longer as safe as possible.

3.4. Modelling of Goal Hierarchy

When developing contradicting SST issues of a SCS it is first necessary to define some initial fundamentals as proposed in Section 3.3.3 to perform MCDM. The approach which has been developed in the context of this thesis analyses the alternative solutions in regards to potentially conflicting SST aspects to calculate the optimal trade-off. For instance, if you analyse the ACC of an automotive vehicle, you might consider different sensor types, data encryption algorithms or data bus systems (see Table 3.2). Hereinafter, it is described how to create the necessary SSTM which is essential for the MCDM. Moreover, it is specified how to create an ADT, an optional extension of the SSTM. Figure 3.4 gives an overview of the used components which are essential for the MCDM.

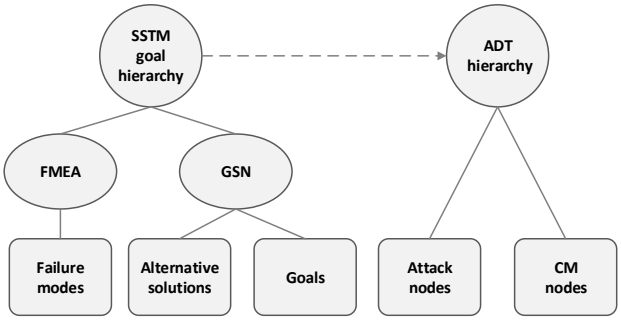


Figure 3.4.: Overview of components of MCDM

3.4.1. SST Model

In the requirements engineering process, it is the aim to define well-defined requirements and to derive solutions from the underlying SM. This knowledge is transferred into a standardised Safety Goal Hierarchy (SGH) which shapes a solid basis for the actual FMEA risk assessment. In this context, the term *safety* of SGH may be misleading since security and timing is taken into account as well. This is explained by the fact that safety is our primary goal which can also be achieved by any security and timing goals, not only by safety goals. The SGH applies the graphical argumentation notation GSN by Tim Kelly and Rob Weaver [KW04] for a comprehensible representation. Therefore, we define a SGH as follows:

Definition 3.2 (Safety Goal Hierarchy)**.** A SGH combines structure of GSN for modelling SST concerns (MC) and FMEA for documenting failure modes.

The GSN presents the goals with various possible features in a standardised, hierarchical and understandable manner. To resolve conflicts of SCSs the concern safety is rated as primary goal whereas security and timing are ranked as sub-ordered goals which might affect safety in an indirect manner. On this account, a

SGH is always initiated with a top-level goal, e.g. *Assuring that the (sub-)system is acceptably safe*. The term *acceptably* means that the system is as safe as possible and ensures to enable almost 100 % of safety. Based on empirical values, exactly 100 % of safety is not reachable. The top-level goal is decomposed into more concrete sub-goals which cover the System under Development (SuD), usually including security requirements and real-time constraints. Figure 3.5 shows a slimmed SGH example of an ACC use case which is extended in Section 3.9. The node *ACC is acceptably safe* represents the primary safety goal which should be achieved. There are four sub-goals in order to fulfil the root goal:

1. *ACC Sensors are working correctly*
2. *ACC Actuators are working correctly*
3. *ACC Software is working correctly*
4. *ACC Communication is acceptably reliable*

Each of them has to be considered for the solutions for which the MCDM should be applied. Goals which are not being refined any more are called Point of Vulnerability (POV). In this example, the solutions refer to the solution set which has been proposed in Table 3.2.

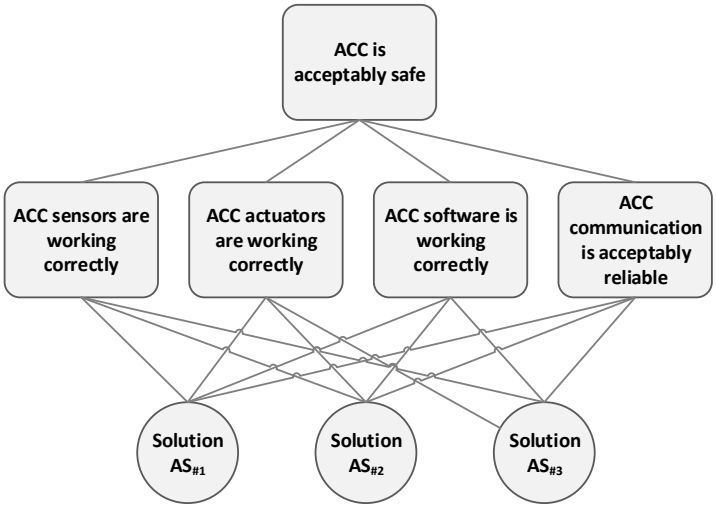


Figure 3.5.: SGH: *ACC is acceptably safe*

3.4.2. Attack and Defence Tree

As already described in Section 2.2 or rather in Figure 2.2 it is the aim to model threats and to develop CMs. The ADT is an extension of the SSTM and provides an additional security analysis considering these aspects. For this purpose, a valid SSTM has to be exist with at least one security goal within the SGH of the SSTM.

Reversely, i.e. there is separate ADT for a specific security goal within the SGH. As already described in Section 2.2.2.1, an ADT provides one or more CMs to mitigate or avoid a security attack. Creating the ADT including its components, i.e. specifying the attacks and developing the corresponding CMs is part of the requirements engineering process. This process is performed in parallel with identifying the *failure modes* which have been explained in Section 3.3.3. First, it is necessary to identify the individual security attacks which can arise. Subsequently, the corresponding CMs can be developed. Figure 3.6 shows a minimalistic example of an ADT hierarchy which is extended in Section 3.9. There is the root attack *Manipulation of the car software* which can be achieved either by the attack *Gain violent access to the car* or by *Gain non-violent access to the car*. In this short example only the first one is explained. The attacks are emphasised by orange nodes. For each attack, possible actions (blue nodes) of the attacker are identified before necessary CMs (green nodes) are developed:

1. Action: *Break down the door* with the CMs
   - *Install alarm system* and
   - *Install security lock*
2. Action: *Exploit vulnerability* with the CM
   - *Install strong authentication mechanism*
3. Action: *Go out unobserved* with the CM
   - *Install a video surveillance equipment*

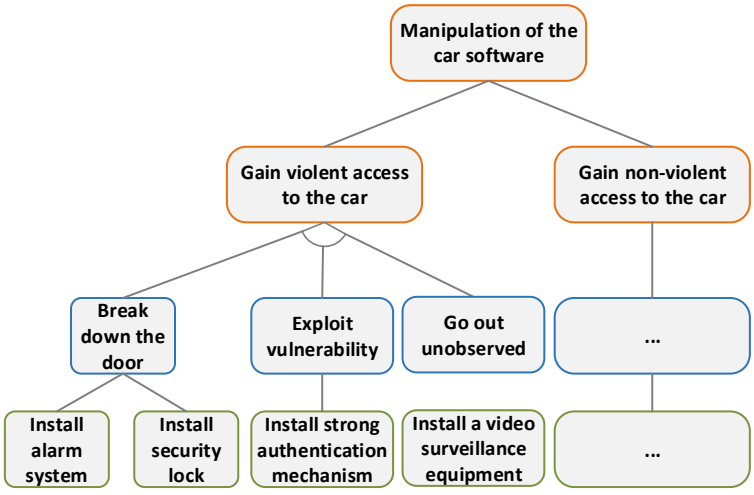


Figure 3.6.: ADT hierarchy: *Manipulation of the car software*

3.5. Applying the AHP on SGHs

It is purpose to develop a well-defined goal structure to avoid or mitigate risks. For this reason, the SGH is introduced. A valid SGH is a suitable model to apply the AHP and thus to perform a MCDM by means of the identified solutions. Some definitions, closely related to graph theory, are introduced. They are necessary for describing the characteristics of a valid SGH precisely.

Definition 3.3 (Validity of SGHs)**.** A SGH is defined as $SGH = (N, E)$ whereas $E \subseteq N \times N :=$ tuple of set of nodes N and edges E. A SGH is valid if and only if it fulfils the following criteria:

1. The number of nodes is finite, i.e. $\exists n \in \mathbb{N} : |N| = n$
2. It defines exactly one root node, i.e. $\exists! r \in N : P(r) = \emptyset$ whereas $P : N \to N$ and $n \mapsto P(n) :=$ set of all predecessors of n
3. The only root node is of the type *GSN Goal*, i.e. $\exists! r \in N : P(r) = \emptyset \wedge Type(r) \in GSNGoal$ whereas $Type : \mathcal{P}(N) \to N$ and $x \mapsto Type(x)$
4. The only root node has at least two children, i.e. $\exists! r \in N : P(r) = \emptyset \wedge |P(r)| \geq 2$
5. Either all children of the root node are of the type *GSN Goal* or *GSN Strategy*, i.e. $\exists! r \in N : P(r) = \emptyset \wedge (Type(r) \in GSNGoal \vee Type(r) \in GSNStrategy)$
6. A child node of the type *GSN Goal* or *GSN Strategy* has at most one parent node, i.e. $\forall n \in N : Type(n) \in GSNGoal \vee Type(n) \in GSNStrategy \Rightarrow P(n) \neq \emptyset$
7. Each node of the type *GSN Strategy* defines at least two children of the type *GSN Goal*, i.e. $\forall n \in N : Type(n) \in GSNStrategy \Rightarrow \exists s_1, s_2 \in S(n) : s_1 \neq s_2 \wedge s_1 \in GSNGoal \wedge s_2 \in GSNGoal$ whereas $S : N \to \mathcal{P}(N)$ and $n \mapsto S(n) :=$ set of all successors of n
8. Each node of the type *GSN Goal* has at least two children of the same type, i.e. $\forall n \in N : Type(n) \in GSNGoal \Rightarrow \exists s_1, s_2 \in S(n) : s_1 \neq s_2 \wedge s_1 \in GSNGoal \wedge s_2 \in GSNGoal$
9. Each node of the type *GSN Goal* has only children of the type *GSN Goal, GSN Strategy or GSN Solution*, i.e. $\forall n \in N : Type(n) \in GSNGoal \Rightarrow S(n) \subseteq GSNGoal \cup GSNStrategy \cup GSNSolution$
10. If a *GSN Goal* G has children of the type *GSN Solution*, then all nodes of the type *GSN Solution* have to be children of G, i.e. $\exists n, m \in N \; \forall s \in GSNSolution : Type(n) \in GSNGoal \wedge Type(m) \in GSNSolution \wedge m \subseteq S(n) \wedge n \in P(s)$

Elements of the type *GSN Context, Assumption* and *Justification* annotate the SGH with useful information which are not mandatory to calculate the optimal trade-off by means of the AHP by ranking the alternative solutions. Therefore, these element types as well as the *in-context-of* relationships can be ignored when checking the validity of SGHs with the aim of determining the best trade-off. For the sake of completeness, the *in-context-of* relationship R within a SGH is defined as follows: $R \subseteq A \times B$ whereas $A \times B = \{(a,b) \mid a \in A,\ b \in B\}$ and $A := GSNGoal \cup GSNStrategy$ and $B := GSNContext \cup GSNAssumption \cup GSNJustification$.

First, the number of nodes has to be finite such that a SGH is valid. This is necessary because violating this requirement would for instance allow endless refining of goals. The SGH is a tree-based structure, i.e. there is exactly one root goal on the top-level. The GSN and thus also the SGH does not provide multiple inheritance, i.e. no *Goal* or *Strategy* node has more than one parent node. In this way, all the *GSN Goal* and *GSN Strategy* nodes forms an hierarchical tree structure by means of the *supported-by* relationships. Furthermore, each *GSN Strategy* node within the SGH needs to be refined by at least two *GSN Goal* nodes. A *GSN Goal* node has to be supported by at least two nodes of the type GSN Goal, Strategy or Solution. The *supported-by* relation R is described formally as follows: $R \subseteq A \times B \cup C \times A$ whereas $A \times B = \{(a,b) \mid a \in A,\ b \in B\}$, $C \times A = \{(c,a) \mid c \in C,\ a \in A\}$, $A := GSNGoal$, $B := A \cup GSNStrategy \cup GSNSolution$ and $C := GSNStrategy$. Since there are only the relationships are available which have been presented in Table 2.10, the only way to terminate refinement process of the goals is to support them by at least two *GSN Solution* nodes. According to Definition 3.3, sub-item 10., each sub-goal must be supported by all solutions which are defined within the hierarchy.

According to rule 3 of Definition 3.3 a *GSN Goal* root node functions as goal with highest abstraction level, i.e. it serves as top-level goal for the SGH. The AHP decision attributes and criteria of this root goal are covered by direct successors of the root node (cf. rule 4 of Definition 3.3). According to rule 5 of Definition 3.3 either a *GSN Goal* or *Strategy* is allowed. These goals or strategies are decomposed further, just like a regular AHP hierarchy. The *GSN Solution* nodes, which are on the lowest level within the hierarchy, match the alternative solutions of the AHP. Applying the AHP algorithm, competing nodes have to be compared pairwise step by step. In the simple example of Figure 3.5 this would mean to judge the relative importance of the goals *ACC sensors are working correctly, ACC actuators are working correctly, ACC software is working correctly* and *ACC communication is acceptably reliable* for accomplishing the top-level goal *ACC is acceptably safe*. Each rating within the AHP should be argued carefully since it may have decisive impacts on the result of the MCDM.

Furthermore, the *solutions* $AS_{\#1-3}$ have to be compared pairwise to determine how well they fulfil the goals *ACC sensors are working correctly, ACC actuators are working correctly, ACC software is working correctly* and *ACC communication is acceptably*

reliable. Since these goals have not been further refined, we speak about Points of Vulnerabilitiy (POVs) and have to be judged according to the FMEA. This means to estimate the probability of occurrence, severity and detection for each of the three solutions individually. This procedure is explained in Section 3.7.1 in more detail. Furthermore, we have to compare how well the solutions $AS_{\#1}$, $AS_{\#2}$ and $AS_{\#3}$ fulfil the objective of the corresponding POVs relatively to each other. Therefore, a pairwise comparison between the individual solutions based on the individual POVs is performed. Finally, the regular AHP algorithm can be applied.

3.6. Improving Consistency of Comparison Matrices

Applying the AHP requires that the matrices of pairwise comparison judgements are consistent (cf. Definition 2.14). In summary this means: If the consistency ratio is above 0.1, the consistency index is more than 10% of the average consistency index of random generated matrices with the same number of columns and rows. Reversely, this means that decision makers have to revise and update their matrices until the ratings achieve a more consistent judgement, i.e. transitivity has to be fulfilled within the whole matrix (cf. Definition 2.13) The bigger the size of the matrices the more difficult it is to achieve a more consistent judgement. If the size of the matrix is greater than 5 it is nearly impossible to create a consistent matrix intuitively. For this reason, an algorithm is proposed how to solve the difficulty of improving inconsistent as well as consistent matrices. The consistency of matrices which have a consistency ratio ≤ 10 can be improved optionally as well.

It is prerequisite that there is a matrix of pairwise ratios A with in/consistent ratios to calculate the completely consistent matrix of pairwise ratios X by means of the maximum right eigenvector v of A, with

$$x_{ij} := \frac{v_i}{v_j}$$

The deviation between them derives a completely consistent matrix X and the matrix of pairwise ratios A can be calculated element by element:

$$A - X := (a_{ij} - \frac{v_i}{v_j})$$

[Saa02] proposed the entry of A with the highest deviation for revision.

Thus, the algorithm for improving the consistency of in/consistent matrices is defined as follows:

Definition 3.4 (Improve Consistency of In/Consistent Matrices)**.** There are four steps necessary to improve consistence of a matrix A consisting of pairwise ratios:

1. Compute the right Perron eigenvector v [LT69] of the matrix of pairwise ratios A.
2. Calculate the fully consistent matrix X with $x_{ij} := \frac{v_i}{v_j}$.
3. Determine the deviations between A and X element-wise: $A - X$.
4. Find the maximum absolute value in the matrix $A - X$.

The algorithm which has been defined in Definition 3.4 is demonstrated by means of a simple example. In this context, there is a matrix A of pairwise comparisons considering three criteria A_1, A_2 and A_3:

$$A = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{bmatrix} 1 & 2 & 4 \\ \frac{1}{2} & 1 & \frac{1}{5} \\ \frac{1}{4} & 5 & 1 \end{bmatrix} \end{array}$$

In this example, A_1 is twice important as A_2 whereas A_1 is four times important as A_3. Consequently, one would imply that A_2 should be twice as important as A_3. However, A_2 has been weighted to be five times less important than A_3 wrongly. In this context we define the following notation (as already used in Section 2.4.1): $A_i \succ A_j$ as A_i is more important than A_j for $i \neq j$. Consequently, we should have $A_1 \succ A_2 \succ A_3$ but also $A_3 \succ A_2$. As can be seen, these are conflicting and inconsistent constraints. According to Definition 3.5, the maximum eigenvalue λ_{max} of A is approximately

$$\lambda_{max} \approx 3,6186$$

Definition 3.5 (Calculation of Maximum Eigenvalue)**.** There has to be a square matrix, e.g. a 3×3 matrix with the following scheme:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Subsequently, the matrix $A - \lambda E$ results by subtracting with λ multiplied unit matrix E [Coh94]:

$$A - \lambda E = \begin{pmatrix} a_{11} - \lambda & a_{12} & a_{13} \\ a_{21} & a_{22} - \lambda & a_{23} \\ a_{31} & a_{32} & a_{33} - \lambda \end{pmatrix}$$

By applying Sarrus' rule [Coh94] the determinant $det(A - \lambda)$ is calculated as follows:

$$det(A - \lambda) = (a_{11} - \lambda) \cdot (a_{22} - \lambda) \cdot (a_{33} - \lambda) + a_{12} \cdot a_{23} \cdot a_{31} + a_{31} \cdot (a_{22} - \lambda) \cdot a_{31} - (a_{11} - \lambda) \cdot a_{23} \cdot a_{32} - a_{12} \cdot a_{21} \cdot (a_{33} - \lambda)$$

The eigenvalues $\lambda_1, ..., \lambda_n$ result from the zeros of the polynomial. The resulting maximum eigenvalue λ_{max} is defined as follows: [Coh94]

$$\lambda_{max} = max(\{\lambda_1, ..., \lambda_n\})$$

According to Definition 2.14 we get a consistency ratio of about

$$CR \approx 0,5869 > 0,1$$

In accordance with Definition 3.4, we first need to determine the right Perron eigenvector of A. It results the following Perron eigenvector:

$$v = \begin{pmatrix} 1.8566 \\ 0.4309 \\ 1.0000 \end{pmatrix}$$

Step 2 of Definition 3.4 requires to calculate the fully consistent matrix X with $x_{ij} := \frac{v_i}{v_j}$. According to step 3 and 4 of Definition 3.4 we need to define the maximum absolute value in the matrix $A - X$. In this way, we receive the following completely consistent matrix X and deviations $A - X$:

$$X \approx \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{bmatrix} 1 & 4,3087 & 1,8566 \\ 0,2321 & 1 & 0,4309 \\ 0,5386 & 2,3207 & 1 \end{bmatrix} \end{array}$$

$$A - X \approx \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{bmatrix} 0 & -2,3087 & 2,1434 \\ 0,2679 & 0 & -0,2309 \\ -0,2886 & 2,6793 & 1 \end{bmatrix} \end{array}$$

If we take a look at the results of the deviation matrix $A - X$ one will realise that there is a maximum absolute deviation value of 2,6793. I.e. the importance of criterion A_3 compared to A_2 as well as the rating of A_2 compared to A_3 should be revised. Changing the proposed rating seems to be reasonable for the example. However, changing the value with the highest inconsistency does not automatically imply that there is an improvement of the consistency index and thus consistency ratio. Appropriately, the algorithm which has been presented in this section has to be repeated until the consistency ratio is acceptable, i.e. less or equal 10%.

3.7. Risk Assessment

The previous sections presented how to model a SGH, applying the AHP on them and how to improve consistency to avoid and mitigate in/consistency. The current section covers step 4 of the concept (cf. Figure 3.1), i.e. the risk assessment is explained in more detail. The MCDM uses the FMEA and ADTA for risk assessment. The second one can be seen as an extension within the SGH and does not affect the result of the MCDM in a direct manner.

3.7.1. FMEA for POVs

In Section 3.4.1 the principle of SGHs has been explained. In this context, the purpose of POVs has been defined. In our approach it is necessary to identify the potential risk of failure for each of the alternative solutions by means of the FMEA. It is shown how rating risks with the FMEA influence the ranking proposed by the adapted AHP. The FMEA is a widely used method to identify and reduce potential risks of failure which are coupled to corresponding systems, processes or hardware, preventatively. [LYL16] First of all, it is recommended to answer the following questions to identify the potential failure modes:

1. What can go wrong?
2. Why did this failure occur?
3. What would be the impact of the identified failure?

If all the potential failure modes including their causes have been identified, they are integrated into the SGH which has been prepared in Section 3.4.1. It may happen that the SGH needs to be refined after identifying the potential failure modes and the causes. Conversely, this entails that the SGH has to be refined until each potential failure mode is described by at least one POV (cf. Figure 3.7).
It is noted that the SGH can be arbitrarily nested and complex, but it is always ending at the POVs, i.e. the leaves of the hierarchical tree structure. If all the mentioned steps have been considered, the potential risks associated with each POV have to be evaluated. When performing a MCDM and respecting SST aspects,

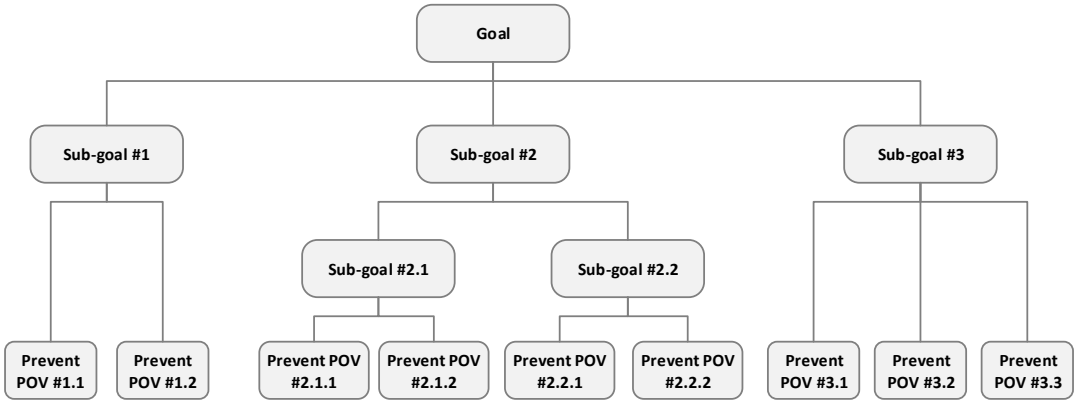


Figure 3.7.: Abstract and exemplary SGH

it is necessary to take the FMEA ratings or the respective RPN of the alternative solutions into account. For this purpose, the FMEA determines the risk based on three parameters which is subsequently referred to as *Occurrence, Severity and Detection (OSD)*:

1. **O**ccurrence: The probability that a failure occurs.
2. **S**everity: The impact that a failure can have it is not detected.
3. **D**etection: The likelihood that a failure will be detected.

Each of these three criteria is assigned an integer value between 1 and 10 whereas 1 means that there is very low probability that a failure occurs. There are also little effects which a failure can have if it is not detected. Moreover, the likelihood that a failure is detected is very high (vice versa for 10, cf. Section 2.2.1.1). The corresponding ratings and how to interpret them can be taken from Section 2.2.1.1. As described in Section 2.2.1.1, to evaluate the risk of the POV, the OSD is integrated in the RPN by multiplying the individual factors: $RPN = O \times S \times D$. Therefore, it can be necessary to consider the RPN of the alternative solutions when performing a MCDM on the SGH. As stated in Section 2.2.1.1, the RPN is clustered into four risk levels. If the RPN does exceed a risk value of 50 or more, retaliatory actions in consideration of risk mitigation or risk reduction are needed. I.e., amendments on the system design have to be done and potentially hazards have to be ruled out, i.e. step 1 to 3 of the concept picture (cf. Figure 3.1) and thus SGH has to be repeated and adapted until an acceptable risk is reached. These changes might cause new threats or existing threats must be updated. In short, this means that this process needs to be repeated until an acceptable risk level and safe state is reached. [BMI17] It is important to note that this approach can not guarantee that the best alternative regarding safety is chosen, because optimising the RPN does not necessarily imply reducing the overall risks. In Section 3.9 a more sophisticated aggregation is presented.

3.7.2. Attack and Defence Tree Analysis

In Section 3.4.2 it has been explained how to create hierarchies based on ADTs. In this context, it has been shown how to develop CMs to mitigate or avoid threats. However, the efficiency of such CMs has not been considered so far. For this purpose, the economic indices Return on Invest (ROI) and Return on Attack (ROA) are introduced to evaluate the efficiency of developed CMs. Furthermore, it is described how to combine the indices to make decisions for CMs.

Return on Invest

It is purpose of the ROI index to measure whether a developed CM is economically profitable with respect to a certain attack from point of the defender. This applies if the ROI value is greater than zero. Otherwise, an investment is not profitable. According to [BFP06] there are some terms which should be understood before proceeding with the calculation of the actual ROI calculation. Hereinafter, these terms are clarified:

- **AV:** The *Asset Value* reflects costs of manufacturing, development, support, renewal and ownership of an asset.
- **EF:** The *Exposure Factor* measures loss or impact on the value of an asset if a threat is realised. Thereby, the *EF* is specified in percent with regard to the value of the asset.
- **SLE:** The *Single Loss Exposure* measures losses of a company if a threat is realised. Since all the threats are not realised with the same probability, this value can be refined by means of an additional calculation of the frequency of the corresponding threat.
- **ALE:** The *Annualised Loss Expectancy* corresponds to the estimated annual loss which suffers a company in case of realising a threat.
- **ARO:** The *Annualised Rate of Occurrence* is a numeric estimation and indicates how often a threat is realised per year.
- **RM:** The *Risk Mitigated* measures how effectively a CM mitigates the risk of a threat. The value is between the interval [0;1].
- **CSI:** The *Cost of Security Investment* is the value for costs which incurs for implementing a corresponding CM.

Based on the given knowledge the algorithm for calculating the ROI index for all CMs of a threat is specified in Definition 3.6.

Definition 3.6 (Return on Invest)**.** If there is an arbitrary ADT in DNF then the ROI index is calculated as follows:

1. Define EF, ARO for all attacks and the AV for the threat.
2. Calculate SLE and ALE for all attacks and assign the results to the corresponding leaves:
$$SLE = AV \cdot EF$$
$$ALE = SLE \cdot ARO$$
3. Define CSI, RM for all CMs.
4. Calculate ROI for all CMs:
$$ROI = \frac{(ALE \cdot RM) - CSI}{CSI}$$

Let us apply a simple example of the automotive domain. We want to calculate the ROI for CM *Install alarm system* which is part of Figure 3.6. According to Definition 3.6 there are four steps which have to be performed. For this purpose, values have to be assigned to some variables which are partly based on experience values.

1. We assign the *AV* to 70.000 €. This corresponds to the prise of the automotive vehicle. In this case, the *EF* is set to 90% since the loss has bad impacts. Furthermore, the *ARO* is set to 0,1. This is due to the fact that this threat is not realised very often.
2. Subsequently, we can continue with calculating *SLE* and *ALE*:
$SLE = 70.000\,€ \cdot 0,9 = 63.000\,€$
$ALE = 63.000\,€ \cdot 0,1 = 6.300\,€$
3. The *RM* of the CM is set to 70%, since a minimal residual risk is present. The *CM* amount to 1.500 €.
4. Finally, the *ROI* can be calculated:
$ROI = \frac{(6.300\,€ \cdot 0,7) - 1.500\,€}{1.500\,€} = 1,94$
In this way, it has been shown that the CM is profitable since the value is greater than zero.

Return on Attack

The ROA index describes the benefit which results from a successful attack over the losses which he or she invests due to the realisation of a CM. According to [BFP06], the calculation of the ROA index uses three parameters which should be described beforehand:

- **GI:** This factor quantifies the expected gain which applies in case of a successful attack for an attacker.

- **Cost:** Estimates the costs for an attack if no CMs are realised.
- **Loss:** Estimates the costs for an attack if the corresponding CM is implemented.

Based on the given parameters the algorithm for calculating the ROA index is specified in Definition 3.7.

Definition 3.7 (Return on Attack)**.** If there is an arbitrary ADT in DNF then the ROA index is calculated as follows:

1. Define GI for the attack.
2. Define Cost and Loss for all attacks.
3. Calculate ROA for all CMs:
$$ROA = \frac{GI}{Cost+Loss}$$

Let us apply the same simple example as used for the ROI but this time from an attacker's perspective. According to Definition 3.7 there are three essential steps necessary:

1. The *GI* for the attack *Break down the door* is set to 50.000 €. This would be the profit for an attacker in case of reselling the automotive vehicle.
2. The costs, which occur if no CM is implemented, would be 2.000 € whereas loss would be 500 € in case of applied CMs.
3. Finally, the ROA for the CM *Install alarm system* can be calculated:
$ROA = \frac{50.000\,€}{1.500\,€} = 20$
Since this value is greater than zero, it is profitable for an attacker.

Analysis with ROI and ROA

It is mandatory to realise at least one CM to mitigate an attack. It is necessary to analyse the indices of each CM together to apply a preferably cost-effective and secure solution which is deterrent for attackers. It is useful to maximise the ROI index whereas the ROA index should be minimised in ideal case. If there is no CM with maximum ROI and minimum ROA index the selection is done either by maximum ROI or by minimum ROA. Alternatively, a pareto-optimal solution can be applied by which the selection is done by means of a customised function. The combined use of ROI and ROA indices enables a more convincing evaluation of CMs. Thereby, not only the benefit but also the deterrent impact on the attacker is considered. [BFP06]

3.8. MCDM Modes

Let us assume we want to find the best ACC system which can be installed within an automotive vehicle. For this purpose, different sensor types, data encryption algorithms as well as the used bus system for data transmission have to be considered. (cf. Table 3.2). However, some of these SST aspects may be conflicting to each other. Therefore, it is necessary to define priorities regarding the individual aspects. The following subsections give an overview in theory how to solve the problem to calculate the optimal trade-off. Therefore, two algorithms, the PCM and RCM, are presented. Thereby, the PCM uses the AHP, whereas the RCM uses the FMEA calculation primarily. In both cases, the result is a percentage distribution of the alternative solutions identifying the one which best fulfils the SST objectives.

3.8.1. Pairwise Comparison Mode

To calculate the trade-off, which is the safest and realisable implementation, Saaty's AHP [Saa04], is applied in a modified version to the SGH, which has already been presented. As already mentioned in Section 2.4.1 the AHP is a MCDM technique which calculates the best compromise on the basis of a hierarchical goal structure and comparison matrices. These matrices contain the relative importances of the individual goals in relation to the corresponding super-ordinated objective. If such a ratio matrix is created, the relative importance has to be ranked for every cohered pair of sub-goals to each other. As proposed by [Saa04] for the AHP, the PCM uses ratings between 1 and 9. Thereby, 1 means that two compared goals are of equal importance, whereas 9 means that goal x is extremely more important than goal y. We have to ensure that the matrix is not inconsistent because of non-transitive comparison ratings. As indicated by [Saa04] the consistency ratio may not exceed 10%. This is due to the fact that comparison matrices with a dimension of 5×5 or more are usually not completely consistent if the matrix is generated by human operators. In Section 3.6 an algorithm has been presented how to improve inconsistency. Besides this algorithm, there are some more similar algorithms in literature which identify inconsistent comparison matrices of the AHP, e.g. [Har87].

By means of the eigenvector and the AHP comparison matrix, local priorities, i.e. the importance of the super-ordinated goal in relation to the sub-goals are calculated. The local priorities are mandatory for the calculation of the final global priority of the POVs, i.e. the absolute importance of the objectives on the leaves for reaching the top-level safety goal is determined. Subsequently, it has to be checked whether the individual alternative solutions are in accordance with the objectives of preventing POVs. This has already been evaluated in context with the FMEA, but with ratings 1 and 10 for failure OSD, therefore FMEA judgements only have to be transferred into suitable AHP comparison matrices which can be done with minimum effort automatically. Since the PCM does not only depend

on the RPN values itself, but rather it depends on the OSD ratings, the PCM allows a more accurate and more flexible way to include FMEA judgements. To avoid the criticism of [Bow03] the OSD of the potential failure modes are rated are individually with respect to the relative importance for the vulnerability of a failure. Otherwise, a RPN value with low risk is possible although one of the OSD values has a high criticality of a failure. For this purpose, a so-called *OSD matrix* is created which is valid for all concerned POVs. This OSD matrix compares the importance of OSD on behalf of safety. In a standard way, the matrix considers OSD equally important, i.e. all the parameters are assigned to 1. Subsequently, it is necessary to transform the FMEA ratings for the alternative solutions into judgement matrices. The individual values of OSD are multiplied with each other, therefore higher ratings result in exponentially higher RPNs.

To consider that characteristic in the matrices of pairwise ratios, a cubic function is needed to transform OSD ratings: Let X be one of the ratings $X \in \{O, S, D\}$ and $x_i \in \{o_i, s_i, d_i\}$ for $i \in \{1, 2, ..., n\}$, then each rating is transformed by $x_i^{'} = x_i^3$ for all x_i. The reason for raising the rating to the third potency is that it approximates the multiplication which would have been yielded when calculating the RPN. Again the inverse ratios of the transformed judgements have to be used, since higher OSD ratings represent a higher risk of failure and though a worse, i.e. a lower, AHP judgement. The three pairwise ratio matrices are calculated as follows:

$$\begin{array}{c} \\ S_1 \\ S_2 \\ \vdots \\ S_n \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & \cdots & S_n \end{array} \\ \begin{bmatrix} 1 & \frac{x_2'}{x_1'} & \cdots & \frac{x_n'}{x_1'} \\ \frac{x_1'}{x_2'} & 1 & \cdots & \frac{x_n'}{x_2'} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_1'}{x_n'} & \frac{x_2'}{x_n'} & \cdots & 1 \end{bmatrix} \end{array}$$

As it can be seen, this is a consistent reciprocal matrix. To calculate the local and global priorities with the AHP, the SGH has been extended for each POV, as depicted in Figure 3.8. The regular AHP algorithm can be applied on this extended SGH to find the safest trade-off between evaluated alternative solutions.

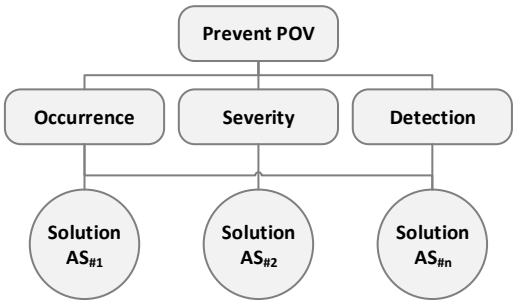


Figure 3.8.: PCM: SGH extension

The following example demonstrates how the local priorities of three alternative solutions S_1, S_2 and S_3 regarding one POV are calculated.

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} O & S & D \end{array} \\ \begin{bmatrix} 1 & 2 & 8 \\ 5 & 8 & 10 \\ 10 & 3 & 1 \end{bmatrix} \end{array}$$

After applying the transformation from the formula $x_i' = x_i^3$ for all x_i we get the following matrix:

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} O & S & D \end{array} \\ \begin{bmatrix} 1 & 8 & 512 \\ 125 & 512 & 1000 \\ 1000 & 27 & 1 \end{bmatrix} \end{array}$$

Subsequently, the matrices of pairwise ratios, for the three criteria OSD and the corresponding priorities are:

$$\begin{array}{c} O \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & Prio. \end{array} \\ \left[\begin{array}{ccc|c} 1 & 125 & 1000 & 0,3487 \\ \frac{1}{125} & 1 & 8 & 0,5789 \\ \frac{1}{1000} & \frac{1}{8} & 1 & 0,0724 \end{array}\right] \end{array}$$

$$\begin{array}{c} S \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & Prio. \end{array} \\ \left[\begin{array}{ccc|c} 1 & 64 & \frac{27}{8} & 0,9898 \\ \frac{1}{64} & 1 & \frac{27}{512} & 0,0005 \\ \frac{8}{27} & \frac{512}{27} & 1 & 0,0097 \end{array}\right] \end{array}$$

$$\begin{array}{c} D \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{cccc} S_1 & S_2 & S_3 & Prio. \end{array} \\ \left[\begin{array}{ccc|c} 1 & \frac{125}{64} & \frac{1}{512} & 0,113 \\ \frac{64}{125} & 1 & \frac{1}{1000} & 0,0010 \\ 512 & 1000 & 1 & 0,9878 \end{array}\right] \end{array}$$

Finally, the priorities for the three solutions can be calculated as follows :

- $S_1 : \frac{1}{3} \cdot (0,3487 + 0,9898 + 0,0113) \approx 0,4499 \Rightarrow$ **44,99 %**
- $S_2 : \frac{1}{3} \cdot (0,5789 + 0,0005 + 0,0010) \approx 0,1935 \Rightarrow$ **19,35 %**
- $S_3 : \frac{1}{3} \cdot (0,0724 + 0,0097 + 0,9878) \approx 0,3566 \Rightarrow$ **35,66 %**

3.8.2. RPN Comparison Mode

The RCM needs a vector $a = (r_1, r_2, ..., r_n)$ of RPNs which can be derived from the following matrix:

$$\begin{array}{c} \\ S_1 \\ S_2 \\ \vdots \\ S_n \end{array} \begin{array}{c} \begin{array}{cccc} O & S & D & RPN \end{array} \\ \left[\begin{array}{ccc|c} o_1 & s_1 & d_1 & r_1 \\ o_2 & s_2 & d_2 & r_2 \\ \vdots & \vdots & \vdots & \vdots \\ o_n & s_n & d_n & r_n \end{array}\right] \end{array}$$

Based on vector a, an inverse vector $a^{-1} = (r_1^{-1}, r_2^{-1}, ..., r_n^{-1})$ is needed. Subsequently, a^{-1} has to be normalised by applying the following formula: $\alpha = \frac{1}{\sum_{i=1}^{i=n} r_i^{-1}}$. The local priority $\pi(S_i)$ for every alternative solution S_i is calculated as follows: $\pi(S_i) = r_i^{-1} \cdot \alpha$. The main difference between the RCM and the PCM is that the RCM directly reflects the RPN values in the MCDM whereas the PCM is much more related to the AHP algorithm by Thomas L. Saaty [SK90].

The following example demonstrates how the local priorities of three alternative solutions S_1, S_2 and S_3 for a POV are calculated, when setting the FMEA ratings for a failure and the three solutions as follows:

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_n \end{array} \begin{array}{c} \begin{array}{cccc} O & S & D & RPN \end{array} \\ \left[\begin{array}{ccc|c} 1 & 2 & 8 & 16 \\ 5 & 8 & 10 & 400 \\ 10 & 3 & 1 & 30 \end{array}\right] \end{array}$$

This means $a = (16, 400, 30)$ and thus $a^{-1} = (\frac{1}{16}, \frac{1}{400}, \frac{1}{30})$ with a normalisation factor of $\alpha = \frac{1}{\frac{1}{16}+\frac{1}{400}+\frac{1}{30}} \approx 10,1695$. The following local priorities for the three alternative solutions result:

- $S_1 : \frac{1}{16} \cdot 10,1695 \approx 0,6356 \Rightarrow$ **63,56 %**
- $S_2 : \frac{1}{400} \cdot 10,1695 \approx 0,0254 \Rightarrow$ **2,54 %**
- $S_3 : \frac{1}{30} \cdot 10,1695 \approx 0,3390 \Rightarrow$ **33,90 %**

When facilitating further reduction and extension of the impact, the rating of the relative importance of the FMEA attributes OSD is enabled. Therefore, the PCM is more adaptable. Since the FMEA supposes an unweighted multiplication of the factors of the rating set for OSD, the assumption has been made that all criteria are equally important. This can easily be adjusted by modifying the OSD comparison matrix A_{RPN}. For instance, one might assess that the probability of occurrence is more important than the worst case severity which is moderately more important than the probability of detecting a failure, i.e. the following applies: $O \succ S \succ D$.

$$A_{RPN} = \begin{array}{c} \\ O \\ S \\ D \end{array} \begin{array}{c} \begin{array}{ccc} O & S & D \end{array} \\ \begin{bmatrix} 1 & 2 & 5 \\ \frac{1}{2} & 1 & 3 \\ \frac{1}{5} & \frac{1}{3} & 1 \end{bmatrix} \end{array}$$

The consistency ratio of A_{RPN} of about 0,35 % is close enough to consistency. The resulting local priorities of the FMEA attributes are:

- Occurrence: 58,2 %
- Severity: 30,9 %
- Detection: 10,9 %

The local priorities of the three solutions would change significantly:

- $S_1 : 0,582 \cdot (0,3487 + 0,9898 + 0,113) \approx 0,786 \Rightarrow$ **78,60 %**
- $S_2 : 0,309 \cdot (0,5789 + 0,0005 + 0,0010) \approx 0,179 \Rightarrow$ **17,90 %**
- $S_3 : 0,109 \cdot (0,0724 + 0,0097 + 0,9878) \approx 0,117 \Rightarrow$ **11,70 %**

3.8.3. Comparison between PCM and RCM

The RCM method and the two versions of the PCM method differ significantly. However, each of them has its own scope. The RCM directly includes the RPN which is an essential part of the FMEA process and can be a compulsory certification requirement. As shown by [Bow03], comparing RPN values is not an easy task. The PCM circumvents this problem by applying the established AHP algorithm while rating alternative solutions relatively to each other in consideration of the three relevant criteria OSD. In particular, this method is useful if the compared alternative solutions have identical RPN which arise from different ratings. The RCM would assess those alternative solutions equally, whereas the PCM prefers the alternative solutions with evenly distributed ratings. Let us demonstrate it with a simple example by means of two solutions with the same RPNs.

	O	**S**	**D**	**RPN**	**RCM**	**PCM**
Solution $AS_{\#1}$	10	7	2	140	50,0 %	41,5 %
Solution $AS_{\#2}$	5	4	7	140	50,0 %	58,5 %

Table 3.3.: FMEA rating: Solutions with equal RPNs

As can be seen in Table 3.3 both solutions have a RPN of 140. Solution $AS_{\#1}$ has a very bad rating for the probability of occurrence of the failure. I.e., there is an increased probability of a very high impact the failure could cause. However, the failure is considered to be inevitably detected. In contrast to that, there is only a medium rating for the probability of occurrence and impact for solution $AS_{\#2}$ but

can not be detected without making contact to the customer. Obviously, the ranking of both solutions is equal when applying the RCM. The PCM ranks the second alternative solutions – with the more equally distributed ratings – slightly better on the assumption that the FMEA attributes OSD have equal importance. Usually, human decision makers prefer equally rated alternative solutions. Therefore, in such cases the PCM would be the better one. The second version of the PCM introduced an even more advanced and customised judgement of the relative importance of the FMEA criteria OSD. However, it has to be indicated that the PCM does not directly improve the RPN. In summary, unless the RPN should be optimised, in most cases it is recommended to use the PCM with a customised OSD rating.

3.9. Example

The example of this chapter analyses two ACC systems which fulfil the necessary safety requirements and enables a cruising speed of at least 160 km/h. The first step is to define and recapitulate the requirements. In this thesis, the functional requirements are available in structural form: *ACC sensors must work correctly*, *ACC software must work correctly*, *ACC actuators must work correctly* and *ACC communication must be reliable*. Based on these requirements, a systematic failure analysis which is based on the FMEA as well as the corresponding SST goals have to be developed. For this purpose, the failure modes and goals are clustered into four categories: ACC sensors, ACC software, ACC actuators and ACC communication. These failure modes are described hereinafter:

1. ACC sensors
    a) Missing an obstacle
    b) Detecting an obstacle not in time
    c) Detecting a non existing obstacle
    d) Distance to the obstacle is not accurate (too short)
2. ACC software
    a) Results are not correct
    b) Results are calculated not in time
    c) Software not secure against manipulation and hacking attacks
3. ACC actuator
    a) Brake fails
    b) Engine fails
4. ACC communication
    a) Communication is not secure against data theft

b) Communication is not secure against manipulation

c) Messages don't arrive in time

d) Messages are transferred incorrectly

Failure 2c) could also be applied for the ACC sensors, actuators and communication. Due to similar properties it is considered only once. Based on these failure modes, it is mandatory to define goals to avoid or mitigate the failure modes. The corresponding goals can be taken from Figure 3.11. Furthermore, it is required to design a SM according to the requirements.

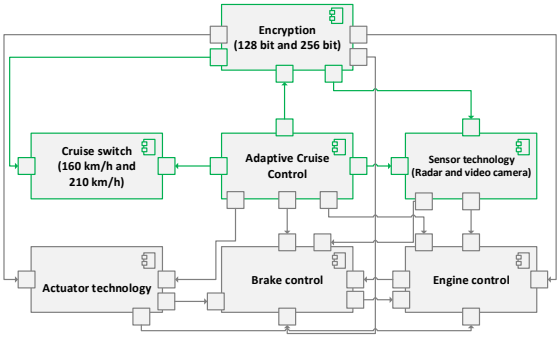


Figure 3.9.: SM of the example

Figure 3.9 represents the corresponding SM. Following the green forks starting from the individual ACC component, the alternative solutions can be retraced:

1. *ACC 160 km/h*: Radar sensor and 128 bit encryption
2. *ACC 210 km/h*: Radar sensor + video camera and 256 bit encryption

The next step covers creating the SGH including goals, POVs and alternative solutions which is depicted in Figure 3.11. The relation of how to use the aforementioned SM within the SGH is illustrated in Figure 3.10.

All sub-goals are split in at least two POVs. For instance, in case of sub-goal *ACC actuators are working correctly* there are two POVs: *Brake failure is sufficiently mitigated* as well as *Engine failure is sufficiently mitigated*. The entire goals as well as all associated POVs (marked with the individual concerns) are represented by means of a SGH where each of the POVs is connected with the two aforementioned alternative solutions. For each POV, it is mandatory to perform FMEA risk assessments as described in Section 3.7.1, i.e. RPN values must be calculated based on OSD probabilities. The exemplary POV *Missing an obstacle can be ruled out with sufficient certainty* with the resulting RPNs is assessed in Table 3.4.

All detailed RPN values can be found in Figure 3.11 assigned to the respective POV. Since there are POVs with associated RPN values, which endanger safety risk significantly, improvements according RPN calculations have to be performed

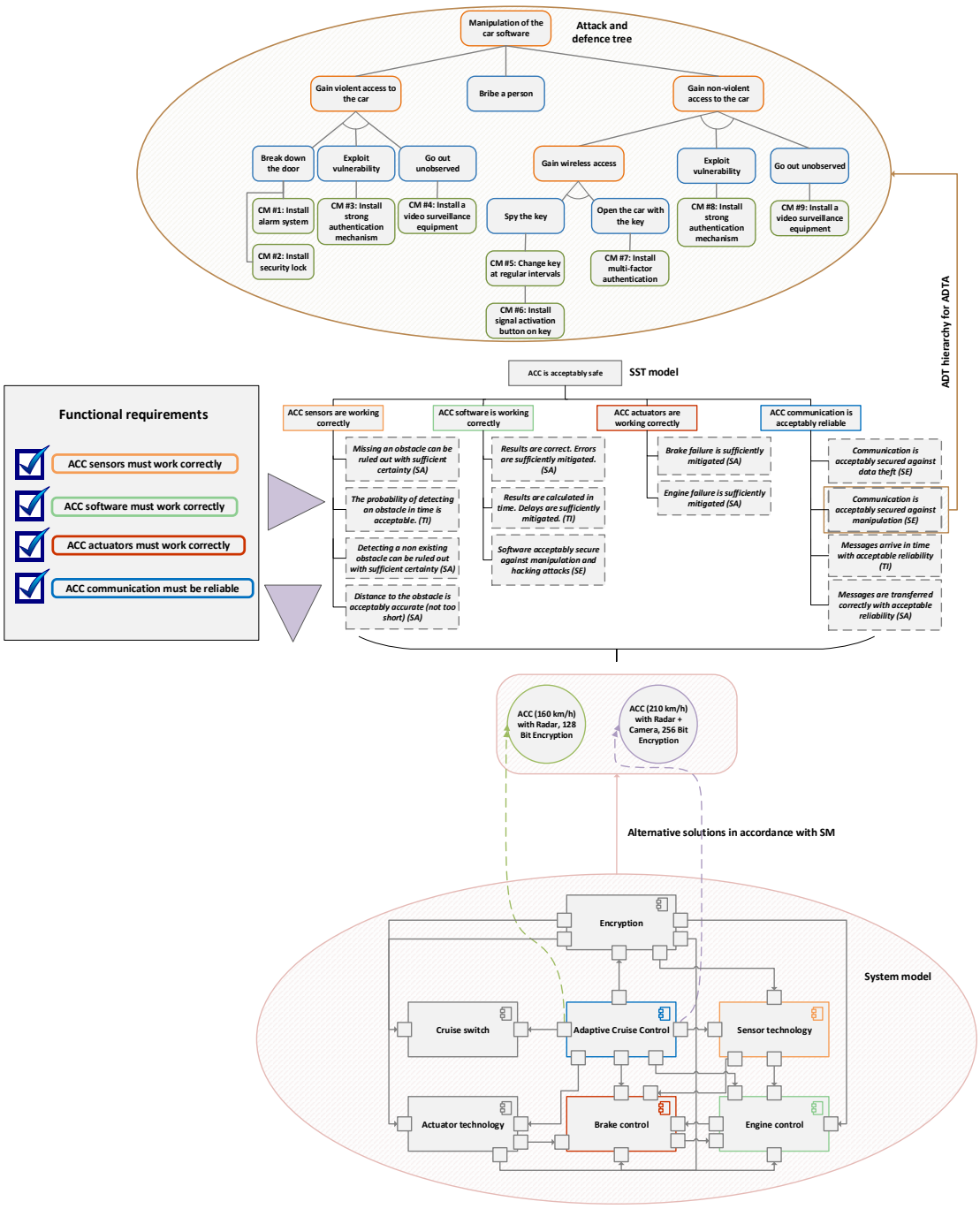


Figure 3.10.: Relations between SGH, SM and ADT

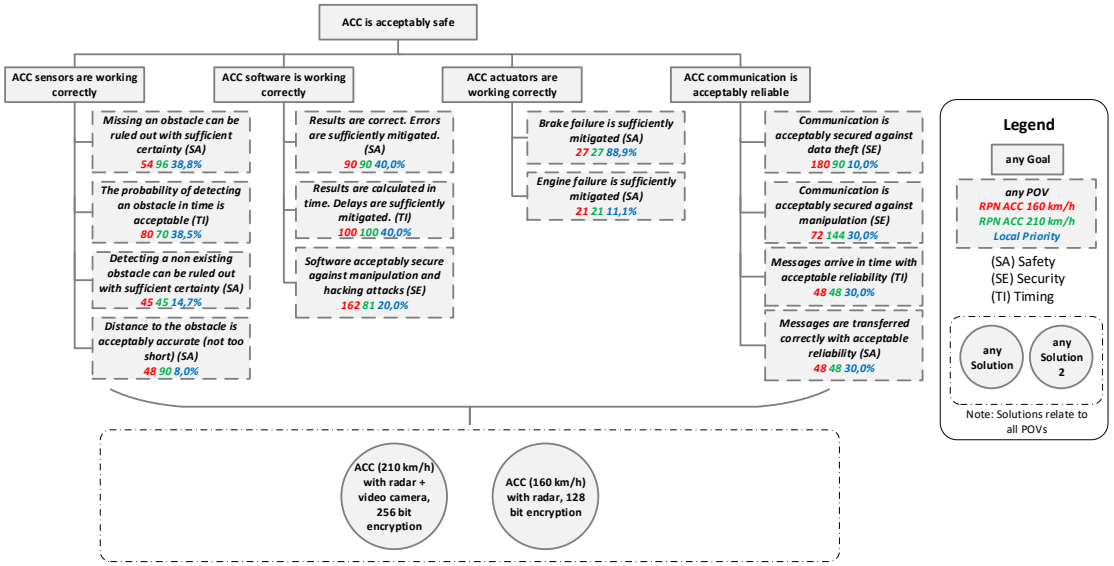


Figure 3.11.: SGH of the example *ACC is acceptably safe*

	O	S	D	RPN
ACC 210 km/h	1	9	6	54
ACC 160 km/h	2	8	6	96

Table 3.4.: FMEA risk assessment of an exemplary POV

but are not further considered in this example. As already mentioned in Section 3.5, it is necessary to rate (sub-)goals and POVs by their importance with the completion of the risk assessments. The AHP rating of the top-level goal can be seen in Table 3.5.

	Actuator	Software	Sensor	Communication
Actuator	1	1	1	$\frac{1}{2}$
Software	1	1	1	$\frac{1}{2}$
Sensor	1	1	1	$\frac{1}{2}$
Communication	2	2	2	1
Local Priority	20 %	20 %	20 %	40 %
Consistency Ratio			0 %	

Table 3.5.: AHP rating of goal *ACC is acceptably safe*

The MCDM can be performed comparing local priorities of the RCM and PCM method. The corresponding results are listed in Table 3.7. For the PCM, the judgements of OSD were changed according to Table 3.6. In this way, the following applies: $O \succ S \succ D$.
The results show that *ACC (210 km/h) with radar + video camera and 256 bit encryption* is safer than *ACC (160 km/h) with radar and 128 bit encryption*. As can be seen, there

	O	S	D	Local Priority
O	1	2	5	58,2 %
S	$\frac{1}{2}$	1	3	30,9 %
D	$\frac{1}{5}$	$\frac{1}{3}$	1	10,9 %

Table 3.6.: OSD matrix of the PCM

is no significant difference between the two methods.

Alternative Solution	RCM	PCM
ACC 210 km/h	**54,4** %	**56,2** %
ACC 160 km/h	45,6 %	43,8 %

Table 3.7.: OSD matrix of the PCM

The SGH hierarchy of Figure 3.11 contains two security goals of which the goal *Software acceptably secure against manipulation and hacking attacks* is examined carefully, i.e. an ADTA is performed. For this purpose, an ADT hierarchy is created first (for the correlation between SGH and ADT, cf. Figure 3.10) which is illustrated in Figure 3.12. Hereinafter, a detailed calculation is performed for the CM

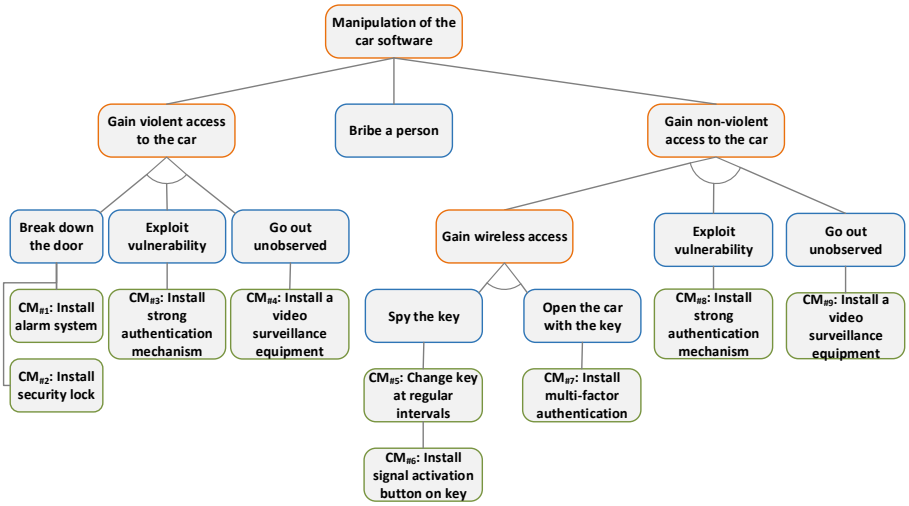


Figure 3.12.: Extended ADT hierarchy: *Manipulation of the car software*

Change key at regular intervals. First, we need to initialise the necessary variables exemplary in order to proceed with the calculation of ROI and ROA:

- **AV:** 40.000 €. In this example, we calculate the ADTA for the manipulation of the car software of a compact car.
- **EF:** The exposure factor is set to 85 %, since (compact-)cars are equipped with a large number of security-critical features. In most cases, the malfunction of these features may lead to almost complete loss of the car.

- **ARO:** Since this kind of car is produced in large quantities, the ARO is set to 10.
- **RM:** This variable is assigned to 0,2 for the CM since it is less effectively.
- **CSI:** Changing the key at regular intervals costs 5.000 €.
- **Cost:** It costs 4.500 € to realise the attack.
- **Loss:** The attack costs 1.500 € if the CM is realised.

According to Definition 3.6, the ROI can be calculated as follows:

$$SLE = 40.000\,€ \cdot 0,85 = 34.000\,€$$
$$ALE = 34.000\,€ \cdot 10,0 = 340.000\,€$$

$$ROI = \frac{(340.000\,€ \cdot 0,2) - 5.000\,€}{5.000\,€} = \mathbf{12,6}$$

Subsequently, the according ROA for the CM can be calculated by applying Definition 3.7:

$$ROA = \frac{40.000\,€}{4.500\,€ + 1.500\,€} = \mathbf{6,67}$$

In practice, it is not possible to realise all available CMs. Therefore, it is necessary to determine the most effective CM. Table 3.8 lists the corresponding values for ROI and ROA according to Definition 3.6 and Definition 3.7. It can thus be concluded that there is no single CM with the best ROI *and* ROA. It is reminded that the ROI has to be be maximised, whereas the ROA needs to be minimised. Therefore, it has to be weighed up in detail which CM applies for more: *Install multi-factor authentication* or *Install strong authentication mechanism*.

CM	#1	#2	#3	#4	#5	#6	#7	#8	#9
ROI	14,20	4,70	6,60	4,43	12,60	11,75	**19,40**	12,60	8,71
ROA	3,08	4,00	1,74	11,76	6,67	7,27	2,05	**1,63**	8,16

Table 3.8.: Results of the ADTA

3.10. Related Work

In this section, related publications and projects are presented and compared with the approach of a MCDM on SCSs. In this context, publications and projects with respect to safety assessment with the AHP, the FMEA, security analysis by means of ADT as well as safety and security in general are considered.

Safety Assessment with the AHP and the FMEA

The AHP by Thomas L. Saaty [SK90] is used for making decisions regarding safety in various domains, e.g. in [Jia+09], [WC11] and [Che+11]. The AHP is also used for making decisions based on security concerns, e.g. in [Ji+10] and [Tah+14]. A MCDM in compliance with safety, considering security and timing as well as functional demands does not seem to have been covered before. Even though the AHP is used for decision making on security concerns, e.g. [Jia+09] and [Tah+14] a MCDM in consideration of safety, taking security and timing issues as well as functional demands into account does not seem to have been researched before. Furthermore, there is the work of [FK18] which combines FMEA, AHP and MULTIMOORA[1] to optimise risk assessment. However, there is no approach which considers MC and calculates an optimal trade-off as proposed in this chapter. Therefore, the approach of this chapter is innovative and does not seem to have been evaluated so far. A fairly similar approach which performs a MCDM in consideration of safety and also combines the FMEA with the AHP, is the work of [ZFW13]. They analyse the reliability of manufacturing processes by means of the Process Failure Mode Effect and Criticality Analysis (PFMECA), enhanced by the well known AHP. This method has solely been designed for analysing safety in manufacturing processes. The method proposed in this chapter can be applied to any SCS, product or process.

Security Assessment with the ADT

There are some publications regarding security assessment, e.g. [HMW11]. This paper proposes an approach how to mitigate or avoid vulnerabilities of database resources and to compare it with agile criteria of a self-defined conceptual model. However, it is only focused on security. In this thesis, we perform security assessment under observation of safety. Furthermore, there are no economic indicators as proposed in the approach of this thesis to measure quality of developed CMs. The approach of [BFP06] has been used in thesis for calculating the economic indicators ROI and ROA based on ADTs. However, in our approach there is an underlying SGH to analyse SST vulnerabilities. Without analysing SST vulnerabilities or security weaknesses a security analysis would not be possible. Moreover, it has been described in this chapter how to update the SGH by using new CMs based on the work of [BFP06]. There are some publications regarding security assessment. It has not been evaluated scientifically before how to integrate the impacts of such an assessment into a MCDM concept.

Related Projects on Safety and Security

The SAFE (**S**afe **A**utomotive so**f**tware archit**e**cture) project which started in July 2011 and has been finished in June 2014 had the aim to "[...] speed up the efficiency development of safety features in cars." [SP14] The focus was to extend and adapt

[1] Multi-objective optimization on the basis of ratio analysis plus full multiplicative form

AUTOSAR[2] system architecture model by methods in to trace safety requirements over the entire project life-cycle [SP14]. Besides safety the SAFE project did not consider any other concerns such as security or timing. Furthermore, there was no research about MCDM and thus no trade-off calculations. The SESAMO (**Se**curity and **Sa**fety **Mo**delling) project focuses on safety and security requirements, aiming "to develop a component-oriented design methodology based upon model-driven technology, jointly addressing safety and security aspects and their interrelation for networked embedded systems in multiple domains." [SES15] One major objective is to develop methods for integrated analysis of safety and security claims which is focused on identifying hazards to facilitate a versed trade-off between conflicting safety and security demands. One goal is to to provide clear evidence, justifying "that the risks associated with the system are as low as reasonably practicable" [Pau+12]. Constituting that a system cannot be safe without considering security demands, the SESAMO project aims to specify safety as top-level goal which can be influenced by security issues. [Pau+12] In contrast to the approach presented in this thesis, the SESAMO project does not provide a competitive MCDM by a systematic method like the modified AHP combined with the FMEA. However, the FMEA is a compulsory part of the certification requirements in the automotive industry [Pau+12]. Moreover, there is another project called SafeCer (**Sa**fety **Cer**tification of Software-Intensive Systems with Reusable Components). It aims to increase "[...] efficiency and reduce(d) time-to-market by composable safety certification of safety certification of safety-relevant embedded systems." [SP16a] It is the aim of the SafeCer project to provide methods and tools composing safety arguments for a complete system by reusing already consolidated safety arguments and proven specifications of the sub-systems. Another project called SYNOPSIS (Safety Analysis for Predictable Software Intensive Systems) which started in 2011 and has been finished in 2016 "is targeting increased efficiency and reduced time-to-market by composable safety certification of safety-relevant embedded systems." [SP16b] The last two projects aim to provide means (architectures, tools, processes or standardisation) and to improve efficient safety assurance and certification. The SYNOPSIS project does not explicitly aim to support a MCDM taking SST issues into account, as it has been proposed in this thesis. [SP16a] [SP16b] Finally, the MERgE (Multi-Concerns Interactions System Engineering) project has to be mentioned. It aimed "to develop and demonstrate innovative concepts and design tools addressing in combination the Safety and Security concerns, targeting the elaboration of effective architectural solutions." [MP17] However, the MERgE project did not take any MCDM algorithms into account, i.e. calculating trade-offs in context of SCSs is not part of [MP17] and thus there is another focus in this thesis. Moreover, there is another huge project with 55 organisations called CESAR (Cost-effective methods and processes for safety relevant embedded systems) [CP12]. It was the aim of that project which started in March 2009 and has been finished in June 2012 to "[...] boost cost-efficiency of embedded system development and safety and certification process[...]" [CP12] in a multi-domain approach. Thereby, the project is concentrated on cost-effective and

[2] Automotive open system architecture

ultra-reliable embedded systems in automotive, aerospace, railway and automation domains since those transportation domains are competitive. This project did not cover an approach for MCDM as proposed in this chapter of the thesis. [CP12]

4
Change Impact Analysis

The preceding chapter described how a MCDM is performed in consideration of SST concerns. In this context, FMEA risk assessment plays an important role and suitable CMs have to be developed possibly if the risk is not acceptable. Introducing new CMs cannot be done offhand since these CMs may have awkward consequences since new risks or hazards might arise. It is part of this chapter to detect dependencies of concerned model components in context of the MCDM and to analyse the extent of the impacts. First, the term *change impact* including a motivation and problem description is clarified in more detail in Section 4.1. The underlying concept is elaborated in Section 4.2 which serves as a basis for the following sections. The impact rules for the algorithms of the change impact analysis are defined in Section 4.3 and its subsections. In this context, the SM, SSTM as well as the ADT are considered. The subsequent section specifies the term *change request* within the software development process and which elements are necessary to define them. To realise such a change impact analysis it is initially essential to define a so-called Change Impact Algorithmic (ChIA) with the necessary algorithms which is part of Section 4.5. For clarification, a plausible example is presented in Section 4.6. Finally, it is conveyed which works are related to our change impact analysis. This chapter is mainly based on the scientific publication [LRB19] of the author and is therefore no cited any more

4.1. Change Impacts

As mentioned in the introducing paragraph of this chapter, the MCDM of Chapter 3 is not a finite approach. The development is progressing steadily, i.e. requirements are constantly changing. For instance, the ADTA may lead to the conclusion that CMs need to be developed and thus some goals within the SSTM have to be changed. In most cases, it is not sufficient to only adapt the affected elements. Rather, other model elements are affected and must therefore also be changed. Hereinafter, the term *change impact* is defined.

Definition 4.1 (Change Impact)**.** A change impact is the result, i.e. effects after changing an existing model element which is caused by a change request. The iterative or recursive process to identify change impacts is called change impact analysis and is an discovery-based approach [Boh02].

In general, a change impact analysis can be performed for single models or even for complex software systems. In both cases, dependencies between individual model elements have to be calculated. Thereby, a distinction is made between

- direct impacts, i.e. there is an effect to one or more neighbour nodes and
- indirect impacts, i.e. effects of direct impacts in turn affect further model elements within the corresponding model or software system

As it can be seen, it is hard to deal with change impacts, especially with indirect impacts since they can trigger ripple-effects. Such effects can already be activated by simple amendments and require reachability information. [LSB15] Let us take an introductory example from the automotive domain: A change request may be like *All data within the bus system must be transferred by means of a 256 bit AES encryption algorithm*. If the current software only supports a 128 bit encryption, a modification of some model elements, e.g., goals within the SSTM is mandatory. Since the change request interferes with a security goal, some further timing goals may be affected by the change request. For instance, the goals *Airbag is triggering in time*, *ACC identifies obstacles in time* and *LDP identifies obstacles in blind spot in time* are directly affected and need be modified as well. This in turn, requires enhancement of safety goals. In case of the goal which is responsible for triggering the airbag in time, some essential safety goals need to be adapted as well. In this way, many more goals may be affected. Another problem is that a target model element can be affected by different source model elements (cf. Figure 4.1) if structural impact rules will be applied. I.e., for a target model element two or more different effect types may be assigned. However, for one change request there has to be exactly one effect type. In Figure 4.1, the change impact analysis is triggered by means of *Model element 1*. The node *Model element 1.2* depends on *Model element 1* and has two different effect types over two iterations. In iteration 1 (green path), *effect A* is assigned to *Model element 1.2*, whereas in the second iteration (blue path) *effect B* is assigned to it. It is also the aim of this chapter, getting this problem under control.

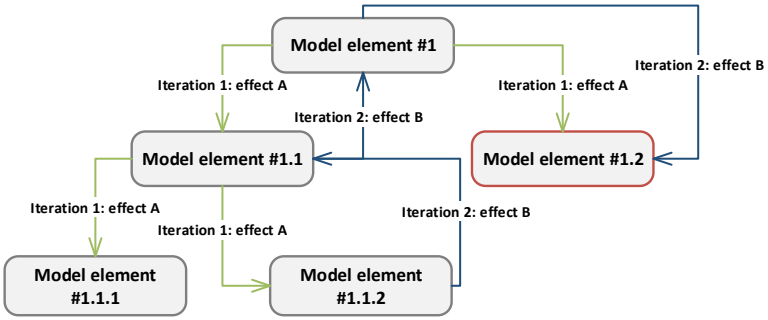


Figure 4.1.: Conflict of the change impact analysis

In this thesis, we distinguish between BC and WC change impacts, i.e. there is minimal set of affected model elements in case of BC change impact analysis.

The WC change impact analysis, however, considers the maximal set of affected elements. In practice, the actual impacts lies somewhere in between and have to be determined by domain engineers. [LSB15] In context of this thesis, we consider four effect types:

1. *No change:* Changing a source element has no effects on the corresponding target model element.
2. *Extend:* Enhancing a source element yields extending target model element. This typically means adding new sub-nodes to the current model element.
3. *Modify:* Amending a source element means modifying target model element. Usually, the context or further information is changed.
4. *Delete:* Changing a source element yields deleting target model element.

4.2. Concept

This section introduces the concept of the change impact analysis by means of a concept picture (cf. Figure 4.2). Thereby, the necessary constituents of the change impact analysis are described. The individual steps of the concept, which are illustrated in Figure 4.2, are explained within this section.

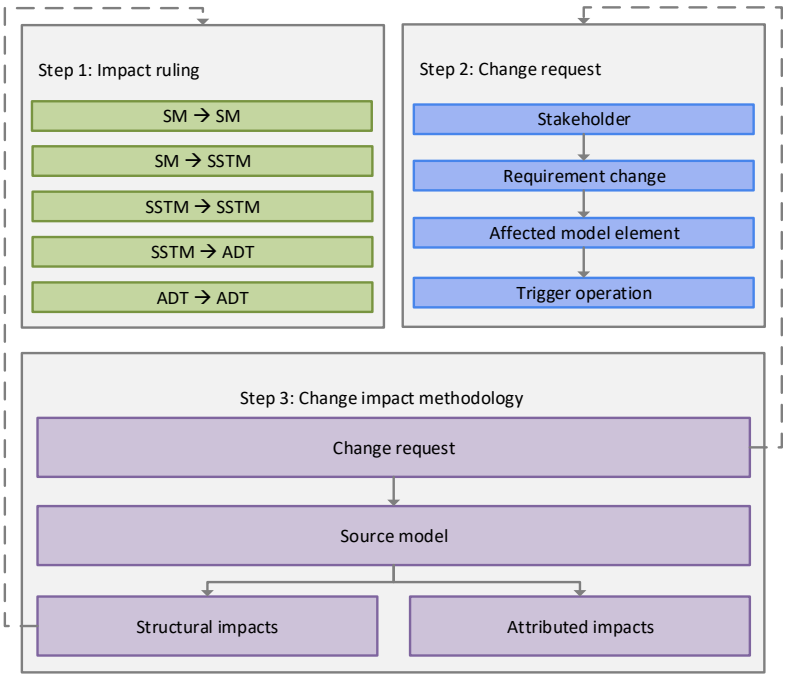


Figure 4.2.: Logical concept of the change impact analysis

When performing a change impact analysis, it is still mandatory to observe SST restrictions as described in Section 2.1. Furthermore, the influences between the

individual concerns (cf. Section 2.1.4) have to be met as well. When realising the results of the change impact analysis, the MCDM as described in Chapter 3 can be performed once again. However, we need to consider and to perform three essential steps which are described hereinafter:

Step 1: To enable an impact analysis, it is necessary to define a heuristic since we usually only know the abstract interrelations of a system but not the causal relationships. In this context, the kind of effects to its correlated (model-)nodes by means of well-defined impact rules will be determined. For each application, there is a set of BC and WC rules respectively. In this chapter, we differentiate between five model dependencies for which the structured impact rules and thus the change impact analysis can be applied. Change requests regarding the SM may either affect the SM itself or the hierarchical SSTM. When modifying the SSTM, it may affect the SSTM itself as well as an ADT. If an ADT is changed it has only impacts to the ADT itself since the ADT is an extension of the SSTM. It is possible that the five model dependencies have to be combined over several iterations. This is part of step 3.

Step 2: As generally described in Section 3.3, change requests consist of requirement change, document management and planning, organisation, controlling which are operated by different stakeholders (e.g. a software developer or safety expert) to get optimal transparency. The term requirement change means that requirements are enhanced (e.g. due to new regulations) and thus a consistent and understandable description of this amendment is needed. In this context, it needs to be determined which model element the requirement change affects, e.g. a goal or POV within the SSTM. Finally, it has to be specified which triggering operation (cf. preceding section) the requirement change is associated. However, the effect type *no change* is not included since the associated model element is the source node (and not any target node). Otherwise, the change impact analysis would not be triggered.

Step 3: The last step of the change impact analysis combines step 1 and step 2 to determine the change impacts. First, we need the change request as defined in the previous step to initiate the change impact analysis starting from a model node with a triggering operation. In this context, the whole source model is used to perform one of the subsequent algorithms. On the one hand, there is a structural algorithm which applies the impact rules of step 1. On the other hand, there is an algorithm which calculates impacts by means of Key Performance Indicator (KPI) based attributions. The first algorithm is applied if there are structurally conditioned change requests whereas the second considers semantics of the corresponding model elements.

4.3. Impact Ruling

When performing change requests, it is distinguished between amendments for which the semantics of qualitative change requests are considered and those for which only the structure is relevant. The latter is part of this section, whereas semantically based impacts are described in further course of this thesis. If the semantics does not matter, only the structure and dependencies of the corresponding models are of interest, i.e. direct and indirect dependencies are considered. In practice, these models are very large and complex and thus structural dependencies are not directly visible. Furthermore, cyclic change impacts can occur and need to be solved accordingly. Let us assume a model element, e.g. a goal within the SSTM, is deleted and we want to determine the structural impacts. In this case, if the impact rules are applied a domain expert receives structural impacts which need to be checked by the expert. As mentioned in the introductory Section 4.1 there are four effect types which may occur: *No effect*, *extend*, *modify* and *delete*. To apply such impact rules to detect structural dependencies, there are five superordinate cases which are described hereinafter shortly:

1. SM $\rightarrow$ SM: When changing system components or relationships between within a SM it may have effects on corresponding model elements of the SM itself.

2. SM $\rightarrow$ SSTM: Since the SSTM requires an underlying SM, amendments regarding the SM have effects on the target SSTM.

3. SSTM $\rightarrow$ SSTM: When modifying the SSTM which is, i.a. significantly responsible for the MCDM may affect the SSTM itself since it compromises a hierarchical structure.

4. SSTM $\rightarrow$ ADT: When enhancing a security goal or POV within the SSTM, which is extended by an ADT, the corresponding ADT is affected.

5. ADT $\rightarrow$ ADT: Similarly to the third item, the ADT is built up hierarchically and thus influences itself.

4.3.1. SM to SM

In this thesis, the MCDM as proposed in Chapter 3 is an essential part. However, to perform such a MCDM an underlying SM is mandatory. If there are change requests, which affect the SM, it may change the SM itself first and foremost. When thinking a couple of steps ahead, these changes may also affect the SSTM and thus influence the result of the MCDM significantly. Table 4.1 lists the impact rules differentiating between BC and WC impact rules for the individual components. According to [LSB15] all the impact rules have the following syntax:

$$\texttt{A.X} \rightarrow \texttt{B.Y}$$

In general, this statement expresses if source A has the characteristics X, it follows that target element B has the characteristics Y. In this concrete use case, this implies A, B $\in$ {*Component*, *DockedPort*, *LinkedPort*} whereas X $\in$ {extend, modify, delete} and Y $\in$ X $\cup$ {noChange} [LSB15]. In further course of this thesis, the elements of X will be abbreviated as {ext, mod, del}. The term component refers to any system component within a SM. The rules SM $\rightarrow$ SM distinguish between docked port and connected Port. To understand the difference between them, the usual order of applying impact rules clearly illustrates port classes. The impact ruling is initiated by a component and the first impact concerns at least one port which belongs to the component from which the impact rule has been started. These part are also called docked ports. Afterwards, one of the rules *Docked Port* $\rightarrow$ *Linked Port* is applied. A linked port is a port which is connected with a docked port. Finally, this linked port has impact/s on the corresponding system component to which this port belongs to.

Source/Target element	**BC**	**WC**
Component $\rightarrow$ Docked Port	A.ext $\rightarrow$ B.mod A.mod $\rightarrow$ B.noChange A.del $\rightarrow$ B.del	A.ext $\rightarrow$ B.mod A.mod $\rightarrow$ B.ext A.del $\rightarrow$ B.del
Docked Port $\rightarrow$ Linked Port	A.ext $\rightarrow$ B.noChange A.mod $\rightarrow$ B.noChange A.del $\rightarrow$ B.noChange	A.ext $\rightarrow$ B.ext A.mod $\rightarrow$ B.mod A.del $\rightarrow$ B.noChange
Linked Port $\rightarrow$ Component	A.ext $\rightarrow$ B.noChange A.mod $\rightarrow$ B.noChange A.del $\rightarrow$ B.ext	A.ext $\rightarrow$ B.mod A.mod $\rightarrow$ B.mod A.del $\rightarrow$ B.del

Table 4.1.: Impact rules of dependencies SM $\rightarrow$ SM

When extending a component, e.g. adding new ports the corresponding docked ports need to be modified both, in BC and WC. In this case, the current docked ports may get new functionality. Otherwise, when modifying a component there is no impact on the corresponding docked port in BC if there are only lightweight changes of the component. However, there are major changes the corresponding docked ports need to be modified since the port need new connections. Furthermore, if a component is deleted the corresponding docked ports are deleted as well in BC and WC since the ports are part of the components. If we regard the docked port as source model element and the docked port is extended then it follows that the corresponding linked ports have not to be changed in BC. In WC, i.e. if the source docked port is of major interest, the corresponding linked ports are extended since the docked port is connected with at least one new connected port. When modifying a lightweight docked port there are no impacts on the connected linked ports. However, in WC they have to be modified as well since the linked port may change its behaviour When deleting a docked port there are no effects on the target linked ports since the linked port is the target port and thus does not have any effects. Finally, it has to be determined which kind of effect it will have for a component if a linked port is changed. If a linked port needs to be

extended, e.g. due to the rule classification *Docked Port* → *Linked Port* it yields no effect in BC. However, in WC the corresponding component needs to be modified. Otherwise, the port could be invalid for the component. Modifying a linked port yields the identical effects as extending them and has also to be justified on the same reasons. If a linked port is deleted, the corresponding component is extended in BC to fulfil same functionality. In WC, the component has to be deleted as well since the component could otherwise no longer exist.

So far, we defined impact rules and the necessity of them. However, it has not yet been explained how to apply the individual impact rules. As already mentioned in Section 4.1, applying different change impact rules may cause some conflicts (cf. Figure 4.1) since there would be different effect types over several iterations for the same model element. To avoid this problem, we assign preferences for BC and WC rules respectively. Hereinafter, the preferences are defined as follows:

$$\texttt{del} \succ_{BC} \texttt{mod} \succ_{BC} \texttt{ext} \text{ and}$$
$$\texttt{ext} \succ_{WC} \texttt{mod} \succ_{WC} \texttt{del}$$

This is due to the fact that it is more complex to extend a model element more as to delete it. In this way, the target model element can change three times maximally. Finally, Algorithm 4.1 specifies how to avoid the above mentioned problems.

Algorithm 4.1 Calculation of impacts of dependencies SM → SM

```
1:  procedure CALCULATIONIMPACTSSM-SM(modelElement, operation)
2:      if modelElement isTypeOf Component then
3:          for all dp ∈ getDockedPorts(component) do
4:              if checkPrefAndApplyRule(component, dp) then
5:                  CalculationImpactsSM-SM(dp, getOperation(dp))
6:              end if
7:          end for
8:      else if modelElement isTypeOf DockedPort then
9:          for all lp ∈ getLinkedPorts(dockedPort) do
10:             if checkPrefAndApplyRule(dockedPort, lp) then
11:                 CalculationImpactsSM-SM(lp, getOperation(lp))
12:             end if
13:         end for
14:     else
15:         for all comp ∈ getComponents(linkedPort) do
16:             if checkPrefAndApplyRule(linkedPort, comp) then
17:                 CalculationImpactsSM-SM(comp, getOperation(comp))
18:             end if
19:         end for
20:     end if
21: end procedure
```

The method *CalculatingImpactsSM-SM* is called by means of two parameters. The first one refers to the corresponding source model element whereas the second

one matches the initial operation (*ext*, *mod*, *del*). First, we need to check our source model element type which is done in line 2, 8 and 16. If the model type is determined it is necessary to fetch all target model elements. Subsequently, for each pair of (source, target) model element the preferences need to be reviewed when applying the impact rule of the current iteration. If the corresponding method *checkPrefAndApplyRule* returns *true* the effect has a higher preference and the current impact rule is applied. Afterwards, the method is called once again. Due to the preferences the impacts can be changed three times in spite of recursion, i.e. it results the following complexity for all model elements $|V|$ and connections $|E|$ between them:

$$3 \cdot (|V| + |E|) \in \mathcal{O}(|V| + |E|)$$

4.3.2. SM to SSTM

To perform a MCDM, it is mandatory to create a well-defined SSTM. The SSTM consists of at least two alternative solutions to get reasonable result. However, if there are change requests which affect the SM, the SSTM is affected as well. In this context, the system components of the corresponding SM have direct effects on the alternative solutions. Since ports of the SM influence components of the SM there are only impact rules for the classification *Component* $\rightarrow$ *Alternative Solution*. In concrete terms, i.e. there are only impact rules with the syntax `A.X` $\rightarrow$ `B.Y` where `A` $\in \{Component\}$, `B` $\in \{AlternativeSolution\}$ and `X, Y` $\in \{ext, mod, del\}$ [LSB15]. Alternative solutions in turn, have no further impact rules since they are the final analysis elements of the MCDM. Hereinafter, Table 4.2 specifies the necessary impact rules differentiated by BC and WC.

Source/Target element	**BC**	**WC**
Component $\rightarrow$ Alternative Solution	`A.ext` $\rightarrow$ `B.mod`	`A.ext` $\rightarrow$ `B.mod`
	`A.mod` $\rightarrow$ `B.noChange`	`A.mod` $\rightarrow$ `B.mod`
	`A.del` $\rightarrow$ `B.del`	`A.del` $\rightarrow$ `B.del`

Table 4.2.: Impact rules of dependencies SM $\rightarrow$ SSTM

When extending a system component of a SM, e.g. by adding new ports, the corresponding alternative solutions within the SSTM need to be modified in BC and WC. The reason is that additional system components have to be taken into account. Obviously, they have to be included into the individual alternative solutions. If a component is modified, it has no effects on the corresponding alternative solutions in BC if there are lightweight changes, e.g. that a component is replaced by another component which is identical in construction. If there are major changes such as enhancing functionality of the component an alternative solution needs be modified since alternative solutions depend on functionality of the components. If there is necessity of deleting a system component the matched alternative solution is modified in that context in WC. Only in BC the alternative solutions is deleted as well since in this way the number of consequential impacts

is kept slightly. As already mentioned in the section before, conflicts can arise when applying different impact rules. Therefore, we define preferences for impact rules of type SM $\rightarrow$ SSTM hereinafter:

$$\texttt{mod} \succ_{BC} \texttt{ext} \succ_{BC} \texttt{del} \text{ and}$$
$$\texttt{ext} \succ_{WC} \texttt{mod} \succ_{WC} \texttt{del}$$

Deleting an alternative solution has lowest priority since results of the MCDM are more significant if there are more alternative solutions within the SSTM. Otherwise it is more easier to modify an alternative solution than to extend it since extending implies applying new innovations. For this reason, the BC ranks *modify* better than *extend* and vice versa for WC. Hereinafter, the underlying algorithm which describes how the impact rules have to be applied:

Algorithm 4.2 Calculation of impacts of dependencies SM $\rightarrow$ SSTM

```
1: procedure CALCULATIONIMPACTSSM-SSTM(component, operation)
2:     for all as ∈ getAssociatedAlternativeSolutions(component) do
3:         if checkPref(component, as) then
4:             applyRule(component, as)
5:         end if
6:     end for
7: end procedure
```

Since we want to calculate the impacts for dependencies between two different model types it is first necessary to fetch all potential alternative solutions which have been assigned to the component. Initially, the calculation method *getAssociatedAlternativeSolutions* is started by means of that component. Subsequently, it is necessary to check the above defined preferences. If the method *checkPref* returns *true*, the corresponding impact rule can be applied. When applying impact rules of type *Component* $\rightarrow$ *Alternative Solutions* no cycles can occur because there is no impact rule starting from an alternative solution. Nevertheless, preferences have to be checked since there may be preceding impact rules which parallelly call the described method of Algorithm 4.2. Finally, we want to analyse complexity of Algorithm 4.2. Since this algorithm is never called recursively a linear complexity of $\mathcal{O}(|V|)$ can be achieved.

4.3.3. SSTM to SSTM

The SSTM which is structured hierarchically is a central component of the MCDM. If the SSTM is changed by external influences, e.g. introduction of new policies it may have impacts on model elements within the SSTM itself. When applying corresponding rules with the syntax scheme `A.X` $\rightarrow$ `B.Y` of [LSB15], which have been listed in Table 4.3, it must be distinguished between three rule types: *Goal* $\rightarrow$ *Goal/POV*, *Goal/POV* $\rightarrow$ *Goal* and *POV* $\rightarrow$ *Alternative Solution*. In summary, there are impact rules which calculate the effects either top-down ($\downarrow$) or bottom-up ($\uparrow$). However, bottom-up rules are more lightweight than the top-down rules.

Furthermore, there are no impact rules starting with alternative solutions since final results of the MCDM depend on alternative solutions.

Source/Target element	BC	WC
Goal → Goal/POV (↓)	A.ext → B.mod A.mod → B.noChange A.del → B.del	A.ext → B.ext A.mod → B.mod A.del → B.del
Goal/POV → Goal (↑)	A.ext → B.mod A.mod → B.noChange A.del → B.ext	A.ext → B.mod A.mod → B.mod A.del → B.mod
POV → Alternative Solution	A.ext → B.noChange A.mod → B.noChange A.del → B.noChange	A.ext → B.mod A.mod → B.mod A.del → B.mod

Table 4.3.: Impact rules of dependencies SSTM → SSTM

When changing a goal within the SSTM there are effects on sub-ordinated goals or POVs. If a source goal is extended the corresponding target goals or POVs need either to be modified in BC or extended in WC. Extending a goal means that a goal is refined, hence the target goal or POV must be modified, i.e. adapted. Only in WC this goal or POV needs to be extended as well since a further hierarchical structuring is necessary. When modifying a goal node there is no effect on the target element in BC since the modifications are more or less negligible. If there are more serious amendments of the source goal, the target node needs to be modified as well. In another case, if a source goal has to be deleted due to change requests it automatically yields deleting the corresponding target goals or POVs both, in BC and WC. The reason for deleting the target node is that it depends on the source goal and thus a delete operation is mandatory. Furthermore, there are impact rules which propagandise effects bottom-up. These include *Goal/POV → Goal*. When extending a source goal or POV then it yields modifying the super-ordinated goal since refining of a node means that the corresponding parent node needs to be updated accordingly. Furthermore, if a goal or POV is modified there are no effects in BC if there are only marginal enhancements. On the contrary, the target node needs to be modified in WC if there are modifications of greater significance. In case of deleting a source goal or POV, the corresponding target goal needs either to be extended or modified depending on choosing BC or WC calculations. In this use case, extending is less sophisticated than modifying. This can be justified by the fact that it is easier to add new functionality than to update an existing one. Finally, changing a POV may have effects on alternative solutions. In BC, i.e. if there are only marginal amendments of the corresponding POVs there are no effects on the target alternative solutions. In WC they need to be modified since the primary goal of safety cannot be guaranteed otherwise by means of the MCDM.

To avoid potential conflicts by applying the impact rules listed in Table 4.3 we define preferences for the BC and WC impact analysis:

$$\texttt{del} \succ_{BC} \texttt{mod} \succ_{BC} \texttt{ext} \text{ and}$$
$$\texttt{del} \succ_{WC} \texttt{ext} \succ_{WC} \texttt{mod}$$

In both cases, BC and WC, deleting a model node is the easiest way since the SSTM has an hierarchical structure and thus information in most cases only is passed along the corresponding paths. Furthermore, in BC it is easier to modify a model element than to extend it since extending requires a few more steps within the SSTM. For instance, if a goal is extended by an additional POV, the POV has to be taken into account for all valid alternative solutions. This is also the reason why extending is better ranked for WC impact analysis and yields better results where applicable.

Algorithm 4.3 Calculation of impacts of dependencies SSTM → SSTM

```
1:  procedure CALCULATIONIMPACTS-SSTM-SSTM(modelElement, operation)
2:      if modelElement isTypeOf Goal then
3:          for all sg ∈ getSubGoalsPOVs(goal) do
4:              if checkPrefAndApplyRule(goal, sg) then
5:                  CalculationImpactsSSTM-SSTM(sg, getOperation(sg))
6:              end if
7:          end for
8:          Goal g ← getParentGoal(goal)
9:          if checkPrefAndApplyRule(goal, g) then
10:             CalculationImpactsSSTM-SSTM(g, getOperation(g))
11:         end if
12:     else
13:         Goal g ← getParentGoal(pov)
14:         if checkPrefAndApplyRule(pov, g) then
15:             CalculationImpactsSSTM-SSTM(g, getOperation(g))
16:         end if
17:         for all as ∈ getAlternativeSolutions(pov) do
18:             if checkPref(pov, as) then
19:                 applyRule(pov, as)
20:             end if
21:         end for
22:     end if
23: end procedure
```

All the impact rules, which have been defined within this, section have to be applied usefully. Therefore, Algorithm 4.3 specifies the appropriate procedure. First, we need to request for the source model element, i.e. it is checked whether a goal or POV performs further actions. This is done by line 2 for *Goal* and by line 12 for *POV*. If the used model element is of type Goal, we need to determine all sub-goals or POVs. For each of them we first need to check preferences as defined in the paragraph above. If the preference check is positive, the corresponding impact rule of type *Goal* → *Goal/POV* is applied and the current method

CalculationImpactsSSTM-SSTM is called recursively. Furthermore, the parent node needs to be determined and the same procedure is repeated, i.e. impact rule of type *Goal/POV* $\rightarrow$ *Goal* is applied. In this way, we can avoid conflicts for cycles over several iterations. In case of model element type POV we first need to determine the parent goal bottom-up and then we check preferences. Afterwards, the corresponding impact rule of type *Goal/POV* can be applied and the current method is called once again. Finally, we need to perform impact rules for type *POV* $\rightarrow$ *Alternative Solution*. Therefore, all alternative solutions have to be determined for each POV. After preference checking the corresponding impact rules can be applied. When determining complexity of this algorithm one will realise that there are as many calculation steps are necessary as for the algorithm of SM $\rightarrow$ SM. The reason is that the impacts of the individual nodes (goals, POVs or alternative solutions) can be changed three times due to assigned preferences. In this context, the recursive algorithm is called repeatedly. This results the following complexity for all model elements $|V|$ and connections $|E|$ between them:

$$3 \cdot (|V| + |E|) \in \mathcal{O}(|V| + |E|)$$

4.3.4. SSTM to ADT

The SGH within the SSTM considers, i.a., security goals to calculate an optimal trade-off. As already mentioned in Section 3.4.2, the SSTM can be extended by an ADT to perform an ADTA and thus to develop appropriate CMs for security lacks. These security lacks are prevented by corresponding security goals within the SGH. If there are change requests, which affect the SSTM, the ADT may be affected as well. In this context, the security goals directly influence corresponding root nodes of the ADTs and thus a few steps further other nodes of the ADT. However, in this section we only consider impact rules of type *Security Goal* $\rightarrow$ *ADT Root Node* to determine impacts between those models. Impact rules which influence ADT elements between them are part of Section 4.3.5. This means in particular that the impact rules with the already familiar syntax scheme `A.X` $\rightarrow$ `B.Y` is composed of `A` $\in \{Goal\}$, `B` $\in \{ADTRootNode\}$ and `X, Y` $\in \{ext, mod, del\}$ [LSB15]. Table 4.4 lists the individual impact rules which are sorted by BC and WC.

Source/Target element	**BC**	**WC**
Security Goal $\rightarrow$ ADT Root Node	`A.ext` $\rightarrow$ `B.noChange`	`A.ext` $\rightarrow$ `B.ext`
	`A.mod` $\rightarrow$ `B.noChange`	`A.mod` $\rightarrow$ `B.mod`
	`A.del` $\rightarrow$ `B.del`	`A.del` $\rightarrow$ `B.del`

Table 4.4.: Impact rules of dependencies SSTM $\rightarrow$ ADT

When extending, i.e. refining a security goal it has no significant effects on the associated ADT in case of lightweight change requests. Only in WC, if there are more pregnant amendments regarding the security goal, the corresponding

ADT needs to be extended since the information about which the security goal is extended has not yet been considered within the ADT. Furthermore, if a security goal is modified it will have no effects in the corresponding ADT root node in BC. In WC the ADT root node needs to be modified as well. Modifying a security goal may yield that the corresponding ADT is no longer suitable since the ADT defines appropriate CMs for the individual security goals. When deleting a security goal, it results in deleting the ADT root node and thus the entire ADT as well. Without an associated security goal an ADT cannot exist. Although impact rules of type cannot cause cycles it is still possible that conflicts will arise. This is possible if an ADT is influenced by two or more security goals. Consequently, we also need preferences for impact type *SSTM* $\rightarrow$ *ADT*:

$$\texttt{mod} \succ_{BC} \texttt{ext} \succ_{BC} \texttt{del} \text{ and}$$
$$\texttt{ext} \succ_{WC} \texttt{mod} \succ_{WC} \texttt{del}$$

In both cases, deleting an ADT root node would be the solution with minimum effort. Conversely, it means that a new adequate ADT needs to be created. In this case, the effort would be greater than to extend to modify it. When extending an ADT there are some more steps necessary as for deleting since new points of attack have to be taken into account. For this reason modifying is rated better than extending in BC and vice versa for WC.

Algorithm 4.4 Calculation of impacts of dependencies SSTM $\rightarrow$ ADT

```
1: procedure CALCULATIONIMPACTS-SSTM-ADT(securityGoal, operation)
2:     ADTRoot root ← getAssociatedADTRoot(securityGoal)
3:     if checkPref(securityGoal, root) then
4:         applyRule(securityGoal, root)
5:     end if
6: end procedure
```

The algorithm which specifies applying impact rules of type *Security Goal* $\rightarrow$ *ADT Root node* is written down in pseudo-code in Algorithm 4.4. The algorithm is called by a security goal. First, it is necessary to determine the associated ADT and thus the corresponding ADT root node. Afterwards, we need to check the preferences regarding the impact rule which should be applied. If the preference is ranked better, the corresponding impact rule can be performed to determine the effect of the ADT root node. Finally, we want to determine complexity of Algorithm 4.4. Since this algorithm does not contain any loops or recursive calls the complexity is constantly, i.e. Algorithm 4.4 $\in \mathcal{O}(1)$.

4.3.5. ADT to ADT

As already mentioned in Section 3.4.2, the MCDM can be extended by a security analysis, in more detail the ADTA which requires a well-defined ADT. For this purpose, the preceding section, covered impact rules of type *SSTM* $\rightarrow$ *ADT* since these security analyses depend on security goals within the SSTM. However, in

this case we only considered corresponding root nodes of the ADTs. We also must take sub-nodes of the ADT into account. Therefore, top-down rules of the type ADT → ADT are needed. Thereby, there are two types of impact rules of the syntactical scheme `A.X` → `B.Y` [LSB15]: *Attacker Node* → *Attacker Node* and *Action Node* → CM. Table 4.5 lists all the necessary impact rules in detail.

Source/Target element	**BC**	**WC**
Attacker Node → Attacker Node	`A.ext` → `B.noChange`	`A.ext` → `B.mod`
	`A.mod` → `B.noChange`	`A.mod` → `B.mod`
	`A.del` → `B.mod`	`A.del` → `B.del`
Action Node → CM	`A.ext` → `B.mod`	`A.ext` → `B.mod`
	`A.mod` → `B.noChange`	`A.mod` → `B.mod`
	`A.del` → `B.del`	`A.del` → `B.del`

Table 4.5.: Impact rules of dependencies ADT → ADT

When extending an attacker node it has no effects on the corresponding target attacker node if there are no marginal refinements. In WC, the target attacker node needs to be modified since there may be enhancements which affects as final result the corresponding CMs. Furthermore, if an attacker node is modified there are no impacts on the corresponding target attacker nodes in BC. In WC, the target nodes need to be modified since modifications, e.g. may enforce other logical operations. For instance, an *AND* branch could be transformed into an *OR* operation branch. If an attacker node is deleted in any step it yields deleting the corresponding target node in BC and WC. This results from the hierarchical structure of the ADT and thus the attacker nodes are included. When extending an action node, which is part of an attacker node, the corresponding CM nodes need to be modified in both cases, BC and WC. This is justified by the fact that extending an action node does not consider the refinement regarding the corresponding CM node. Therefore, it has to be still modified. In another case, if an action node is modified there is no effect on the CM node in BC. However, in WC the corresponding CM node needs to be modified as well. In BC the amendment is not marginal, however, in WC it has serious impacts regarding SST if an inadequate CM is realised. When deleting an action node it yields deleting the corresponding CM nodes since they depend on the action node, i.e. without any action node there is no CM node. If the impact rules regarding the ADT are applied it is possible that conflicts arise. For instance, if there is one CM, which mitigates attacks of two or more action nodes, a conflict may arise when changing all the action nodes over several paths. For this reason we need preferences to avoid conflicts:

$$\texttt{mod} \succ_{BC} \texttt{del} \succ_{BC} \texttt{ext} \text{ and}$$
$$\texttt{mod} \succ_{WC} \texttt{ext} \succ_{WC} \texttt{del}$$

In both cases, modifying has the highest priority since modifying does not entail any structural changes. In BC, deleting is ranked better than extending since the structure is changed in case of extending and thus requires a few more steps. In

WC analysis deleting has the worst priority since it often requires creating a new (sub-)tree afterwards and the effort is not really minimised by this operation. To apply the corresponding impact rules, an procedure is necessary which is specified in Algorithm 4.5.

Algorithm 4.5 Calculation of impacts of dependencies ADT → ADT

```
 1: procedure CALCULATIONIMPACTS-ADT-ADT(modelElement, operation)
 2:     if modelElement isTypeOf AttackerNode then
 3:         for all an ∈ getAttackerNodes(attackerNode) do
 4:             if checkPrefAndApplyRule(attackerNode, an) then
 5:                 CalculationImpacts-ADT-ADT(an, getOperation(an))
 6:             end if
 7:         end for
 8:     else
 9:         for all cm ∈ getCMNodes(counterMeasure) do
10:             if checkPref(actionNode, cm) then
11:                 applyRule(actionNode, cm)
12:             end if
13:         end for
14:     end if
15: end procedure
```

First, we need to query the model element type, i.e. *AttackerNode* or *ActionNode* which is done in line 2 and 8. In case of an attacker node it is subsequently necessary to determine all target attacker nodes. Afterwards, it is checked by means of the preferences if the needful impact rule can be applied. If this has been done successfully, the procedure is called again recursively. In the second case the target CM nodes have to be determined before the preferences can be checked. Subsequently, the corresponding impact rule can be performed. The impact rule *Action Node* → *CM* does not require a repeated recursive call since CM nodes have no outgoing transitions. The complexity of Algorithm 4.5 is $\mathcal{O}(|V| + |E|)$ since there are $|V|$ model elements and $|E|$ connections between them. Furthermore, the algorithm does not contain any cycles and has hierarchical structure.

4.4. Change Requests

Definition 4.1 specified the term *change impact* by using the term *change request*. Hereinafter, the term *change request* is defined.

Definition 4.2 (Change Request)**.** A change request is a defined request for modification which usually receives after determining the extent of the requirements to be realised in a project bindingly [Mäk00].

The more complex a system is, the more likely it is that there are change requests during the project life cycle. Even long project times increase the likelihood of

necessity of change requests. A change request is a uniform defined requirement which usually delivers a principal to a corresponding contractor. [Mäk00] However, according to [Mäk00] there are some more use cases for the necessity of change requests:

1. Errors and failures have been identified by means of bug reports and need to be avoided.
2. Customers or users wish to adapt and to enhance system properties.
3. Errors and failures have been detected while developing a linked or related system.
4. There are some changes in the underlying structure, architecture or standards. For instance, an alternative operating system or safety standard should be used instead.
5. Commands from a higher instance lead to change requests.
6. An individual software or hardware configuration is no longer available on the market.

Thereby, project requirements can be extended, modified or reduced. In principle, to realise a change request, a series of information has to be noticed. First, the part of a product or software which should be changed is described textually. By analogy it is also written down in which state the product or software should be changed. It is mandatory to motivate and justify the amendment. In case of an individual software product, the involved software license or version has to be specified additionally. Moreover, the date of the change request as well as corresponding date have to be indicated. The employee which initiates the change request should mention his or her name in the proposal. Furthermore, the change request should also contain estimations of costs separated by items, e.g. personnel costs or license fees. Moreover, an estimated time frame should be specified separated by useful milestones. [Kaj99] In further course of this thesis, a change request only consists of four essential elements for the sake of simplicity unless otherwise stated:

1. Stakeholder
2. Description of the change request
3. Affected model element
4. Trigger operation

Let us take a simple example of a change request which is illustrated in Figure 4.3. In this example, the encryption should be made more secure. For this reason, we need the stakeholder *Security Expert*. The change request description *Update encryption algorithm from 128 bit AES to 256 bit AES* is assigned to the stakeholder.

In this way, the intention of the change request has been specified in more detail. Afterwards, the description is refined and assigned to the corresponding model element. In this case it refers to the SSTM goal *Software acceptably secure against manipulation and hacking attacks* (cf. Figure 3.11). Finally, the triggering operation (extend, modify or delete) must be selected which can be usually derived from change request description. In this example, the goal is modified.

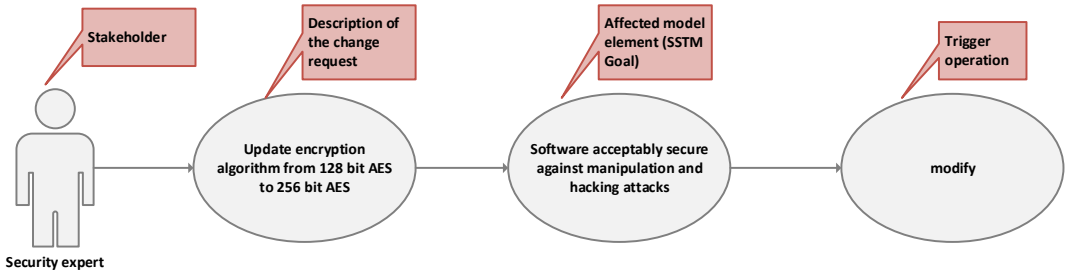


Figure 4.3.: Example of a change request

If change requests are initiated, efficiency is increased due to less subsequent failures or errors and thus low risk of adjustments. Furthermore, by means of change requests costs are minimised since rethought and newly developed systems are much more expensive. Moreover, in this way there is an efficient maintenance since transparency is fulfilled from the change request to realisation of the corresponding change requests. [SAP19]

4.5. Change Impact Algorithmic

So far, it has been specified how impact rules of different source and target model types are performed. Furthermore, the necessity of change requests has been explained in detail. It has not been considered how to combine different impact ruling types as proposed in Section 4.3.1 to Section 4.3.5 to a consistent methodology. This is part of Section 4.5.1. Moreover, there is another methodology which calculates resulting change impacts based on attribution functions. This procedure is presented subsequent in Section 4.5.2.

4.5.1. Structural Impacts

In practice, it is not sufficient to determine only resulting affected elements if a change request is performed. In this context, traceability is of great importance, i.e, the entire path has to be comprehensible to also perform unforeseen modifications. As described in Section 4.3, these kinds of safety-critical aspects do not consider semantics but ensure that structural dependencies are not violated. For instance, if there is a road sign recognition for speed limits the structural impact analysis could calculate that the installed video camera needs to be adapted. In retrospect it might turn out that a change has to be made on the corresponding bus system

which is linked with the video camera. Without a reasonable traceability it would not be possible to detect this component. In this way, it is purpose of structural impacts to check which model elements of all interrelated models are concerned of a change request. This kind of change impact analysis is easier to perform than an attributed and semantic change impact analysis since it does not require further information, e.g. Safety Integrity Level (SIL) classification.

In Section 4.3 it has been described how to apply impact rules of the corresponding types, e.g. *SSTM* $\rightarrow$ *SSTM*. However, in practice there are impacts which not only affect change impacts of one type. Therefore, we need to combine them. Thereby, it is possible that a target model element is affected by impact rules of two different types. For instance, an *alternative solution* could be affected by the impact types *SSTM* $\rightarrow$ *SSTM* as well as *SM* $\rightarrow$ *SSTM*. For this reason, we need to specify preferences which are listed in Table 4.6. In case of SM as target model, there is no alternative as source model. However, if a SSTM model element is affected the impacts of the SSTM itself have more expressiveness than the SM since the SSTM is more essential for the MCDM. In other words, impacts of the type *SSTM* $\rightarrow$ *SSTM* are ranked better than *SM* $\rightarrow$ *SSTM* and thus have priority. If an ADT model element should be changed by means of the impact analysis it can either be changed by means of the ADT or the SSTM. In this case, the ADT is ranked better since it contains necessary security analysis components.

Target model	**Preferences source model**
SM	SM
SSTM	SSTM $\succ$ SM
ADT	ADT $\succ$ SSTM

Table 4.6.: Preferences of structural impacts

Finally, all gathered information is collated to a procedure which is specified in Algorithm 4.6. First, we need to determine the model element from which the structural impact analysis is initiated. This is done by method *getAffectedModelElement(changeReq)*. Subsequently, the preferences which have been listed in Table 4.6 can be applied. If the target model element is part of the SM, the change impact rules as described in Section 4.3.1 can be performed. Otherwise, i.e. in case of SSTM or ADT model element the preferences must be checked if there would be conflicts. Afterwards, the corresponding impact rules of types as described in Section 4.3.2 to Section 4.3.5 can be applied.

4.5.2. KPI Based Impacts

It is usual that some quality attributes and SST properties are met. These include, e.g., compliance with corresponding SILs or timing restrictions. For this purpose, we want to find those model elements within the entire models (SM, SSTM and ADT) which have similar SST properties (e.g. identical SIL classification) than the model element from which the change impact analysis will be triggered. This

Algorithm 4.6 Calculation of structural impacts

```
1:  procedure STRUCTURALIMPACTS(changeReq)
2:      modelElement ← getAffectedModelElement(changeReq)
3:      if getTarget(modelElement) isTypeOf SM then
4:          applySMImpactRule
5:      else
6:          if checkPreferences then
7:              applyPrioritisedImpactRule
8:          end if
9:      end if
10: end procedure
```

procedure only considers KPI, i.e. attribution without any implicit structural dependencies. In practice, there are often semantic dependencies even though the concerned model elements do not correlate structurally. For instance, if a system component of a SM is replaced by another system component which boasts a different SIL it may affect goals within a SSTM. To enable attribution, i.e. KPIs for model nodes, we need to define construction of them which is introduced in Definition 4.3.

Definition 4.3 (Attribution)**.** For each node there is at least one triple which consists of following characteristics:

1. *Attribute key*: Defines attribute which is taken into account, e.g. *SIL classification* for safety concern.
2. *Attribute function*: Defines a function with which allowed values are specified. In this thesis there are three types of attribute functions:
   a) *Max*: Specifies that a maximum defined threshold may not be exceeded. This function is, e.g. usually applied for timing functions.
   b) *Min*: Defines that a corresponding minimum may not be undercut. This function is, e.g. applied if at least a specific SIL must be achieved for a model element.
   c) *Exactly*: A precisely defined value has to be maintained, i.e. there is exactly one valid value. This function is, e.g. applied.
3. *Attribute threshold*: Defines a limiting value which limits the corresponding attribute functions.

For instance, if there is a model element, e.g. a goal regarding timing concern there may be an attribution triple with the following parameters: *(transmissionTime, max, 300)*. In this context, transmission time is taken into account for the corresponding model element. Thereby, transmission may not last more than 300 ms. In addition to the attribution triple there is also a further triple

$$\Delta_{FMEA}(O, S, D)$$

which only affects POVs and is part of the SSTM. This triple is specified within the corresponding change requests and defines the individual deltas of the probability components for calculating RPN, i.e. occurrence, severity and detection. For instance, if there is a $\Delta_{FMEA}(-2, 0, +1)$ it means that occurrence is downgraded two ranks whereas detection is getting better one level and severity remains unchanged if the corresponding change request is realised. In this way, the MCDM can be initiated once again if the change request only affects the SSTM. Furthermore, if the KPI based change impact analysis is triggered it could be the case that there is more than one attribution triple. For the precision of the change impacts it is necessary to define whether all annotated KPIs should be taken into account or whether it is sufficient to consider at least one of them. For instance, if there is model element *Airbag is acceptably safe* which may have two attribution triples, e.g.

- *(timeLimit, max, 150)*
- *(silLevel, min, 2)*

In this case it has to be decided whether it is sufficient to only consider either attribute *timeLimit* or *silLevel*. Otherwise, both of them are taken into account for calculation of affected model elements. To realise this difficulty, a boolean variable *allAttributes* is needed which is set to `true` if all attributes are considered or `false` if not. Some of the triples or variables as proposed in this section have to be taken into account within the change request. These include the *allAttributes* in any case and Δ_{FMEA} if necessary. Since we cannot make any logical predictions about operations, e.g. extend, modify or delete we combine them as a modifying operation. The adaptations of the attributed change requests are illustrated in Figure 4.4.

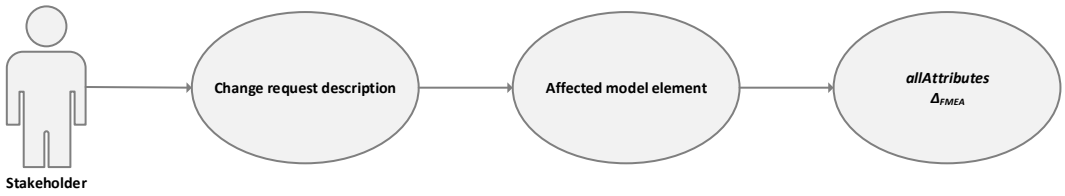


Figure 4.4.: Change request for KPI based impacts

In Algorithm 4.7, the procedure is defined which enables calculating of attributed impacts in consideration of the prerequisites which have been mentioned in the last paragraphs. First, we need to determine the affected model element which is either part of the SM, SSTM or ADT and is specified within the change request. Afterwards, it needs to be checked whether all attribution triples are necessary for the calculation of the change impacts or only one of them respectively. Subsequently, each model, i.e. SM, SSTM and ADT is browsed for vulnerabilities of the corresponding attribution triples by means of the methods *ImpactsAll/OneAttributeXX*. Thereby, *XX* stands for the individual model types. Concretely, this means that all the attribution triples of the model nodes are checked regarding the attribution

triple/s of the triggering model element. In this context, it has to be taken care that the *max, min* or *exactly* function is met. For instance, if there is a system component sc_{old} which has to be replaced by another system component sc_{new}. Thereby, the following applies for the individual system components according to Figure 4.5:

- sc_{old}: (timeLimit, max, 500)
- sc_{new}: (timeLimit, max, 300)

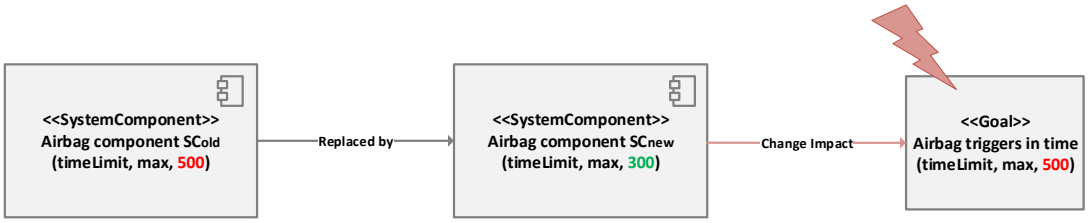


Figure 4.5.: Example of an attributed change impact

In this case, it is looked for model nodes with attribution triples which currently allow a maximum `timeLimit` of more than 300ms and thus violate the attribution triple *(timeLimit, max, 300)* of sc_{new}. This kind of calculations are part of the method *findVulnerabilities(XX)* whereas *XX* stands for the model type, e.g. SM which needs to be browsed. Furthermore, if the affected element is of type POV the new RPN regarding the FMEA can be calculated automatically if the corresponding alternative solutions are not affected by triggering change request. In this way, the MCDM can be can be performed once again without any significant effort. This is done by line 20 to 22. All the affected elements are finally stored in the list *affectedElements* and has been modified during the calculations. The complexity of this algorithm is $\mathcal{O}(|V| + |E|)$ where $|V|$ is the number of all model nodes and $|E|$ the relations between them. This can justified by the fact that each element is inspected at most once.

4.6. Example

The example, which is presented in this section, clarifies the principle of structural and attributed change impact analysis by means of two selected use cases. The use case of the attributed change impact analysis is the already known ACC example from Section 3.9 which is extended by necessary attribution triples. For the structural change impact analysis a new use case is introduced which calculates change impacts for a road sign recognition.

Structural Impacts

The first use case is demonstrated by means of a SSTM (cf. Figure 4.6) which represents a SGH for the goal *Speed limit road sign recognition is acc. safe*. The root

Algorithm 4.7 Calculation of attributed impacts

```
1: procedure ATTRIBUTEDIMPACTS(changeReq, allAttr)
2:     modelElement ← getAffectedModelElement(changeReq)
3:     if allAttr then
4:         for AttributionTriple at ∈ getAttributionTriples(modelElement) do
5:             ImpactsAllAttrSM(at)
6:             ImpactsAllAttrSSTM(at)
7:             ImpactsAllAttrADT(at)
8:         end for
9:     else
10:        for AttributionTriple at ∈ getAttributionTriples(modelElement) do
11:            ImpactsOneAttrSM(at)
12:            ImpactsOneAttrSSTM(at)
13:            ImpactsOneAttrADT(at)
14:        end for
15:    end if
16: end procedure
17: procedure IMPACTSALL/ONEATTRIBUTEXX(attributeTriple)
18:    affectedElements ← new List
19:    findVulnerabilities(XX)
20:    if getAffectedElement(attributeTriple) isTypeOf POV then
21:        calculateNewRPN(getDeltaFMEA(changeReq))
22:    end if
23: end procedure
```

safety goal is refined by three further goals: *Road sign recognition camera is acc. safe*, *Road sign recognition software is acc. safe* and *Road sign recognition communication is acc. safe*. The first one is refined by further safety and timing goals which consider malfunction due to back-lights as well as the correct and timely recognition of road signs. The second and third goal is refined by SST goals which are already known by Section 3.9. Finally, the SSTM contains two alternative solutions which cover two different image sensor types. Only one of them supports uncompressed data storage. First, we need to define a change request for this use case which should be performed:

1. The resolution of cameras has at least 20 mega pixels.
2. It affects the POV *The probability of detecting a road sign in time is acceptable*.
3. This results in the triggering operation `modify` of the POV.

For this change request a structural change impact analysis is performed since the correct recognition of speed limit road signs is minimum safety-critical since the road sign recognition only supports the driver and does not perform interventions. As already indicated, the structural change impact analysis is initiated by the purple dashed goal of Figure 4.6 with `mod`. We have to apply rules of type *SSTM*

→ *SSTM* as proposed in Section 4.3.3 to get the desired results. When performing a BC analysis there are no further impacts. This analysis is chosen if the current camera system approximately fulfils the new requirements with minimal deviations. In WC analysis some changes are needed over several iterations. When applying rules of type *Goal/POV → Goal (↑)* the green bordered goals of Figure 4.6 are affected with `mod`. In the next steps and iterations we apply *Goal → Goal/POV* rules. The POVs which have some `mod` impacts are highlighted with a red dashed frame. Finally, the affected solutions which needs to be `modified`, can be determined when applying the corresponding rule `A.mod → B.mod` of the type *POV → Alternative Solution*. In this way, it has been demonstrated that modifying a single POV can have impacts on the remaining POVs within the SSTM. In this use case it is justified by the fact that the camera shots which are primarily affected by the change request have to be processed by further software. Furthermore, their results need to be communicated within any bus system.

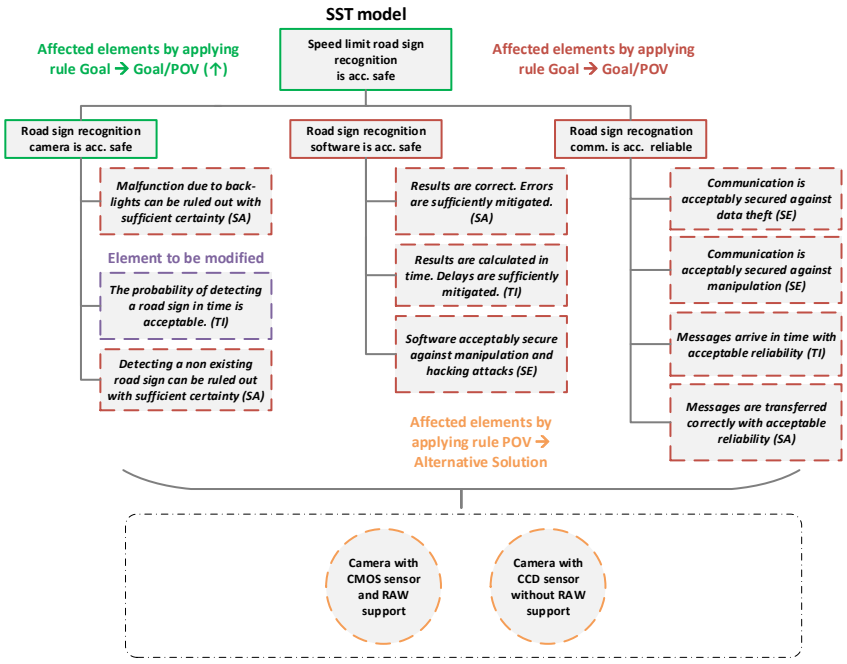


Figure 4.6.: Example of structural impacts

KPI Based Impacts

The use case for calculating attributed impacts is based on the example of Section 3.9. First, we need to take some precautions to perform the attributed change impact analysis which is illustrated graphically in Figure 4.7. The individual POVs of the SSTM are extended by corresponding attribution triples to consider SST. As already mentioned in Section 3.3.2, a SM is required to derive individual alternative solutions therefrom. Therefore, we need to annotate individual selected

system components with attribution triples as well. The sensing must fulfil timing constraints, i.e. there is also an attribution triple with attribution key *timeLimit*.

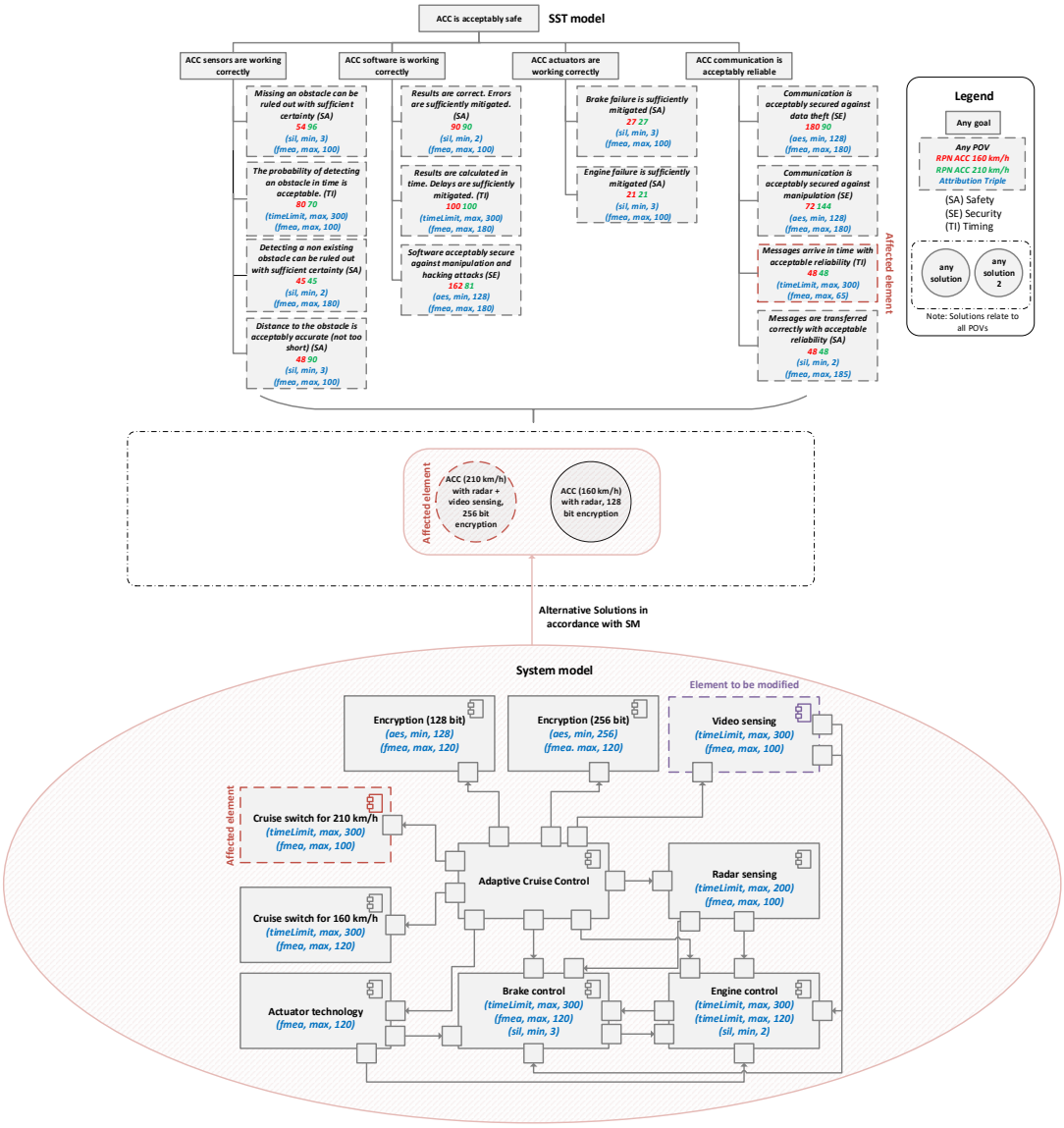


Figure 4.7.: Example of attributed impacts

Moreover, the individual encryption algorithms have specific key lengths. Therefore, we need an attribution triple with the key *aes*. In accordance with the SSTM, the system components are annotated with corresponding *fmea* triples as well. In this use case we define a change request with following characteristics:

1. The system component *Video sensing* is replaced by another one.
2. The new video sensing have some new attribution triples
   a) (timeLimit, max, 400) and

b) (fmea, max, 70).

3. *allAttributes = true*

4. $\Delta_{FMEA} = (0, -1, 0)$

For the safety concern there is the attribution triple *(sil, min, X)* where *X* stands for the corresponding SIL, i.e. at least SIL *X* has to be achieved. Furthermore, there is an attribution *(aes, min, X)* where *X* stands for the key length of the AES algorithm. The attribution triple *(timeLimit, max, X)* is assigned to the timing concern, i.e. there is a maximum time limitation of *X* ms. Furthermore, each POV and some system components are annotated with a triple *(fmea, max, X)*, i.e. the FMEA of the corresponding POV or system component does not exceed a RPN of *X*. The attributed change impact analysis is consequently initiated by system component *Video sensing*. Since we consider all attribution triples we need to check for violations of *timeLimit* and *fmea* for corresponding model nodes. The first model node which is affected by the change request is the system component *Cruise switch for 210 km/h* which uses *Video and radar sensing*. Therefore, the alternative solution *ACC 210 km/h with radar + video sensing, 256 bit encryption* is affected. Furthermore, the POV *Messages arrive in time with acceptable reliability* is affected by the change request since there is a maximum required time limit of 300 ms and FMEA RPN value of 65. Both attributes could be violated by the new video sensing. Since the change impact analysis yields an alternative solution an automated adaption of the corresponding FMEA values and thus an iterated execution of the MCDM is not possible. In summary, there are three affected model elements which have been marked with a red dotted frame in Figure 4.7:

- System component *210 km/h*
- Alternative solution *ACC 210 km/h with radar + video sensing, 256 bit encryption*
- POV *Messages arrive in time with acceptable reliability*

4.7. Related Work

There are some related publications which are presented and compared with the approach of a change impact analysis on SCSs. On the one hand there are papers which cover Enterprise-related impacts. On the other hand there are publications dealing with change impacts regarding UML models and object-orientated programming languages, e.g. Java.

Enterprise Based Impacts

First, the work of [Arn96] has to be mentioned. This publication introduces basic techniques and concepts which are necessary to perform a change impact analysis. In contrast to [Arn96] this thesis proposes an advanced ChIAs. Moreover, it is

not part of [Arn96] taking any safety-critical concerns into account. The work of Langermeier et al. [LSB15] provides a change impact analysis approach based on Enterprise Architecture Models (EAMs). By means of their approach, which is based on a data-flow analysis technique, it is analysed which model elements are affected. Moreover, the algorithm of the authors aims to apply a change impact analysis in context of EAMs. However, in this chapter of the thesis an approach is proposed how to apply change impact analysis taken safety-critical concerns, e.g. SST into account. Furthermore, there is the paper of Hanemann et al. [HSS05] dealing with resource failures which might endanger service level agreements by influencing services. Therefore, [HSS05] presents an approach which identifies the effect of resource failures with respect to the corresponding services and service level agreements. The work of [HSS05] analyses failure impacts and does not take safety-critical issues into account. Finally, there is an impact analysis from [Ola07]. "The analysis consists of stakeholder impact index to determine the nature and impact of stakeholder influence, the probability of stakeholders exercising their influence and each stakeholder's position in relation to the project [...]" [Ola07]. The focus of this work is on impacts of stakeholder influences whereas the focus of this chapter of the thesis is on the calculation of change impacts on the basis of models in safety-critical environment. In summary, this chapter of the thesis proposes an approach how to apply a structural and KPI based change impact analysis taken safety-critical concerns, e.g. safety and security into account. The focus of the related publications is on other domains, e.g. EAM.

Change Impacts of UML Models and Programming Languages

The work of Briand et al. [BLO03] describes the impacts of change requests to UML diagram elements. In this context, potential consequences of a change are identified and it is determined what modifications need to be made to accomplish a change. This work is limited to only UML diagrams and does not allow any other self-defined model types, e.g. SSTM. In contrast to this thesis, the paper of [BLO03] is not focused on safety-critical aspects. The publications of Ren et al. [Ren+04] as well as [RT01] propose an Eclipse based tool with the name Chianti and analyses change impacts of regression or unit tests. By means of the execution behaviour a set of affected changes is determined for each affected test. Chianti does not analyse change impacts on models but on Java code. The work of [Ren+04] and [RT01] enable textual based change impacts in the form of Java code. In this context, it is not the aim of [Ren+04] and [RT01] to enable a maximum degree of SST by means of the change impact analysis. The publication of [Fis+05] presents an software suite called Margrave which verifies and analyses change impacts of access-control policies. In this context, a "semantic differencing information between versions of policies" [Fis+05] is performed. It is noted that there is no attributed change impact analysis as proposed in this chapter of the thesis. Moreover, the reference to SCSs is not given. Therefore, the approach presented in this chapter of the thesis seems to be novel.

5 Multi-Concerns in Software Product Lines

The last main chapter describes application of Multi-Concerns in Software Product Lines. First, the term *Software Product Line* is defined and their necessity is given in Section 5.1. The subsequent Section 5.2 describes the concept of the current chapter including a helpful overview picture in detail. The first essential part of this chapter is covered in Section 5.3.1. Thereby, it is generally described how MCDMs for SPLs have to be realised and performed to get an optimal result for each safety-critical SPL. However, calculating trade-offs is getting more complex since SPLs are getting more extensive. For this reason, an algorithm is specified to reduce complexity and to cluster semantically equivalent features of SPLs. Thereby, structural and KPI based algorithms are used which are specified in detail in Section 5.3.2. Since there may be change requests which concern SPLs the impact ruling of Section 4.3 needs to be extended by four more rule types, namely FM $\rightarrow$ FM, SM $\rightarrow$ FM, SSTM $\rightarrow$ FM and FM $\rightarrow$ SSTM. To internalise content of this chapter, the methodology is exemplified by means of a selected example. Finally, related publications are referenced and the topic of this chapter is delimited from them in Section 5.6. This chapter is mostly based on the authors' work [LB19] and is therefore no further cited.

5.1. Software Product Lines

Nowadays, customers prefer a variability of software products to choose the product which is adapted to one's own needs. In this context we speak about Product Line Engineering (PLE). From the company's point of view it is the aim to meet (SST) requirements of the customers. Furthermore, it is essential to save money, effort and time. Moreover, companies are concerned about reusing their know-how and artefacts [PBD05] This process is divided into two phases as illustrated in Figure 5.1: Domain engineering and application engineering. Thereby, it is purpose of the domain engineering process to specify potential products by defining commonalities and variabilities of the corresponding product family. Application engineering on the other hand defines the methodology of building products by applying specifications of the domain engineering process and to prepare an implementation therefrom. Product variants are defined by selecting features from the underlying FM which represents the SPL in a graphical

notation. Product lines are used in several business and application units. [Sch18] However, this thesis is focused on SPL, i.e. software is built on the basis of PLE.

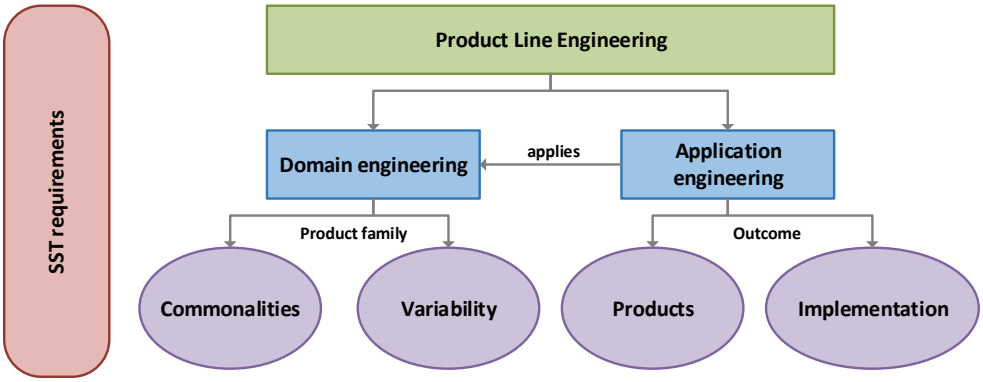


Figure 5.1.: PLE process

Definition 5.1 (Software Product Line)**.** According to Harmann et al. [Har+14] a SPL is specified as a collection of software products with similar properties which vary in several aspects but also have a common basic functionality.

In general, applying SPLs have advantages and disadvantages. On the one hand SPLs increase maintenance since it concerns more than one software product variation usually. Furthermore, applying SPLs enables easier testing under certain circumstances since a set of similar features can be tested in the same test cycle. Moreover, fulfilment of functional requirements is given by modelling of features by means of FMs. On the other hand, due to large number of variability it is not always easy to find an optimal software product configuration. [Har+14] This is also reflected in the context of embedded and SCSs since there is a large number of parameters and it is pretty hard to find the optimal configuration.

In the present thesis SPLs are used to model software-based features of SCSs by means of FMs. Depending on individually selected SPL configurations the MCDM is performed with the advantage that we have not to create a separate SSTM for each SPL configuration. For this reason, only the relevant branches within the hierarchical SSTM are taken into account for calculation of the optimal trade-off.

5.2. Concept

This section describes the concept of applying MC in context of SPLs and is subdivided into two parts which can be read independently of each other. First, it is explained in general terms which steps are necessary to realise a MCDM using SPL techniques. This methodology is extended by a procedure which reduces complexity in a way that not all SPLs have to be analysed. The second part is a

complement of the change impact analysis of Chapter 4 and includes the first part into the approach. The necessary individual steps are illustrated in Figure 5.2.

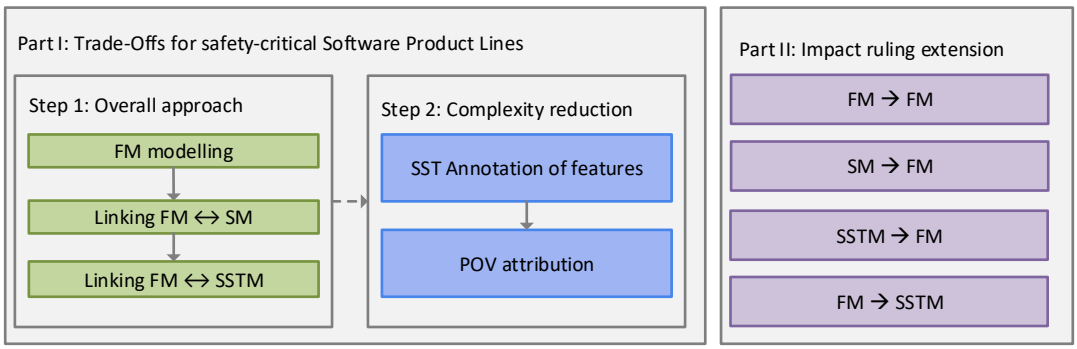


Figure 5.2.: Logical concept of applying MCs in SPLs

When performing a MCDM, it is essential to take necessary steps into account as described in Chapter 3. The same applies when applying a change impact analysis. Hereinafter, it is described which steps are necessary if a MCDM for safety-critical SPLs should be realised. Realising a MCDM for safety-critical SPLs requires an existing SSTM which should be minimised or hide irrelevant elements of them according the corresponding SPL. Furthermore, it is conveyed how impact rules have to be extended to calculate change impacts regarding MC in SPLs.

Part I → Step 1: First, we need a FM which represents all the variabilities and features of the safety-critical SPL. For instance, there may be a FM which represents ADASs of an automotive vehicle. To enable MCDMs for safety-critical SPLs it is essential that there is a SM and SSTM (cf. Section 3.3.2 and Section 3.4.1) as a basis. First, we need to link features of the FM with the underlying system components of the SM. In this way, relevant system components on system level are discovered which will be later transferred on the SSTM. Afterwards, all the features are linked with the corresponding POVs of the SSTM in order to determine necessary POVs for corresponding SPLs. For instance, an ACC SPL configuration may only take ACC POVs into account for the MCDM. This ensures that only relevant POVs are considered for calculation of the optimal trade-off and the results are not falsified.

Part I → Step 2: Step 2 is a continuation of *Part I → Step 1* to reduce complexity for calculating trade-offs of safety-critical SPLs. This step is necessary because conventional SPLs cover a large number of configurations which can not be taken into account completely to get an optimal trade-off. Therefore, it is essential to cluster semantically equivalent features with approximately identical SST requirements. First, all features need to be annotated with SST tags to determine the corresponding safety-critical concerns of the individual features. Secondly, a feature related attribution of POVs within the SSTM is needed to take further safety-critical characteristics into account. These include, e.g. risk assessments

(FMEA calculations).

Part II: In practice, there are permanent change requests of SPLs, i.e. the underlying models need to be enhanced. This means, modifications of the FM yield amendments of the FM itself and the linked SSTM. Besides this, the SM and SSTM may be modified as source model and have effects on the FM as target model. Consequently, there are new change impact rules of type $FM \rightarrow FM$, $SM \rightarrow FM$, $SSTM \rightarrow FM$ and $FM \rightarrow SSTM$. As already described in Section 4.3 the impact rules are differentiated between BC and WC, i.e. there is a minimum and maximum set of effects.

5.3. Trade-Offs for Safety-Critical Software Product Lines

The current section covers Part I of the concept picture which is subdivided into two steps. First, an overall approach is described in detail to realise SPL techniques for calculation of an optimal trade-off. The subsequent and final subsection outlines how to reduce the complexity of them to calculate trade-offs more efficiently.

5.3.1. Overall Approach

In general, the overall approach covers three steps and consists of necessary dependencies between FM, SM and SSTM which are illustrated in Figure 5.3.

Step 1: To realise trade-offs for safety-critical SPLs it is first essential to model functional demands in a standardised and hierarchical manner. For this purpose FMs (cf. Section 2.3.2) have been established to model such functional demands. The individual features are annotated with flags to consider SST concerns. Thereby, features can derive SST annotations from their parent features. Let us assume we want to determine trade-offs regarding ADASs. In this case it is modelled which combinations of different ADASs are permissible and which are not. For instance, if an ACC is included in the current configuration, a simple CC is excluded automatically since an ACC has all functionality of a CC. However, SPL combinations are often restricted by structural measures, e.g. built-in sensor technology. In this way, the MCDM needs to find out whether sensor fusion of a SPL configuration is acceptably safe. It is therefore the aim of SPL based MCDM to inspect whether a SPL configuration is acceptably safe. In this manner, manufacturers can test before market launch profitability of such configurations. Therefore, cost-intensive recalls could be avoided.

Step 2: As already indicated in Figure 1.4 SMs are part of functional demands. Therefore, it is obvious that a FM also needs to be linked with system components (and their subcomponents) of SMs. To calculate trade-offs which have to fulfil a

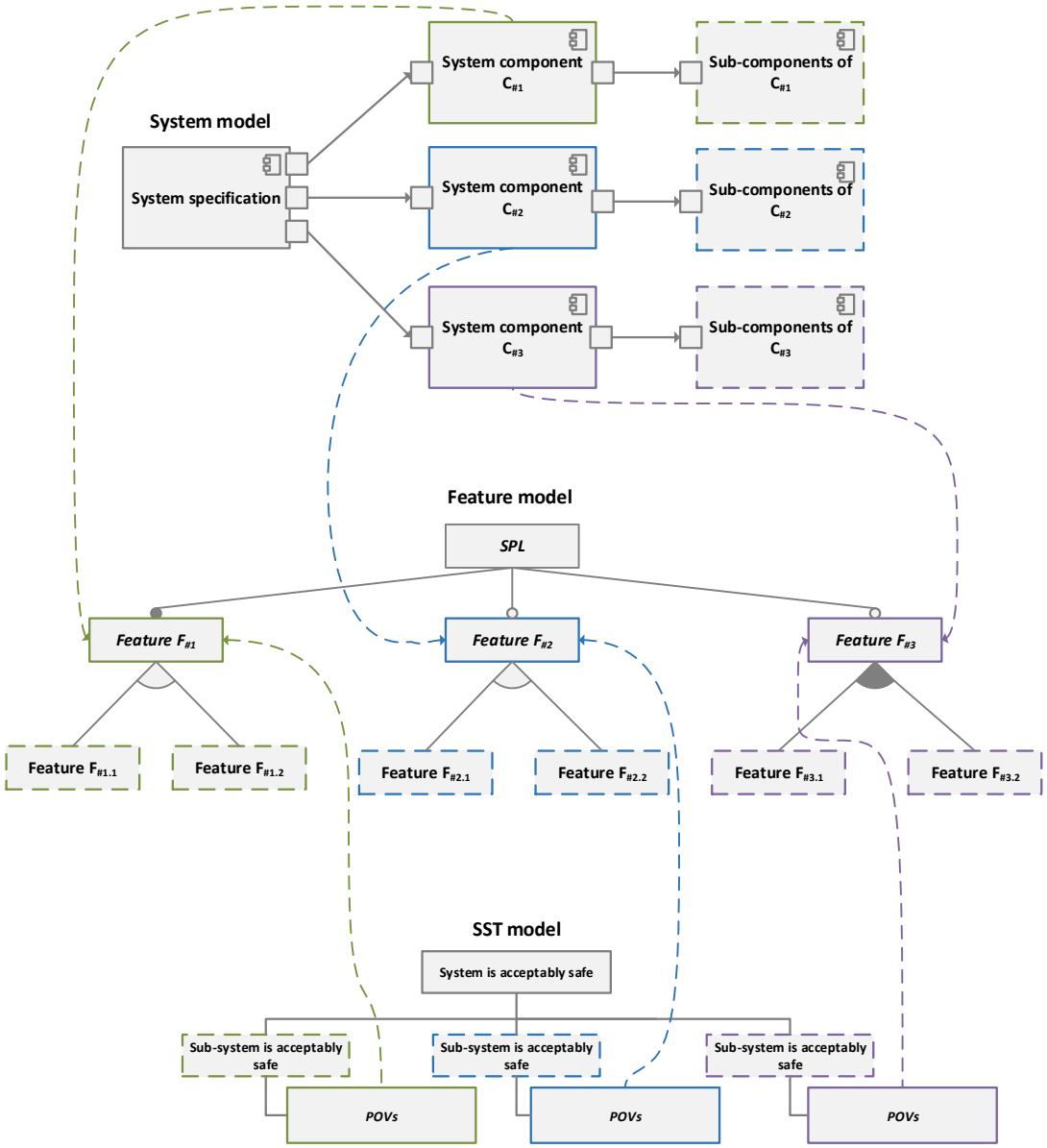


Figure 5.3.: Dependencies between FM, SM and SSTM

certain degree of safety a SSTM is required. As described in Section 3.4.1 a SSTM entails alternative solutions which are derived from an underlying SM. Depending on selected SPL configuration alternative solutions are relevant or not or need to be updated if applicable. In this way, relevant solutions regarding the selected feature path are enabled. Therefore, a linking between FM and SM is mandatory to determine involved system components of a selected SPL configuration. For instance, if there is a feature ACC which can either be equipped with a radar sensor or a camera then alternative solutions referencing on either radar, camera or both are applicable.

Step 3: It is the main objective for calculation of optimal trade-offs to fulfil qualitative requirements, i.e. a linking between FM and SSTM is mandatory. I.e., each relevant feature of the SPL configuration is linked with the corresponding POVs of the underlying SSTM. In this way, the MCDM considers only linked POVs and its associated goals up to the root goal for calculation of trade-offs of individual SPL configurations. According to Definition 2.16 the MCDM is based on an hierarchical approach which calculates local priorities to determine global priorities, i.e. percentage distribution of the corresponding alternative solutions. Since SPL enable customised configurations it is usual to only consider a subset of child-goals or POVs to calculate the individual local priorities. However, disabling a child-goal or POV yields another percentage distribution of the individual child-goals or POVs and thus influences calculating of trade-offs. For instance, if there is matrix A with AHP weights for child-goals A_1, A_2 and A_3 (cf. Table 5.1) with $A_1 \succ A_2 \succ A_3$ the local priorities are

- $A_1 : 75,1\%$
- $A_2 : 17,8\%$
- $A_3 : 7,0\%$

$$A = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{bmatrix} 1 & 5 & 9 \\ \frac{1}{5} & 1 & 3 \\ \frac{1}{9} & \frac{1}{3} & 1 \end{bmatrix} \end{array} \quad A' = \begin{array}{c} \\ A_1 \\ A_2 \end{array} \begin{array}{c} \begin{array}{cc} A_1 & A_2 \end{array} \\ \begin{bmatrix} 1 & 5 \\ \frac{1}{5} & 1 \end{bmatrix} \end{array}$$

Table 5.1.: AHP matrices with weights for $A_1 \succ A_2 \succ A_3$

Let us assume there is a SPL configuration which does not take goal A_3 into account (cf. Table 5.1), the reduced matrix A' has the following local priorities:

- $A_1 : 83,3\%$
- $A_2 : 16,7\%$

When comparing the individual local priorities A_1 and A_2 it is noticeable that percentage distribution for A_1 better results than for A_2. It still applies $A_1 \succ A_2$,

i.e. there is no further scaling necessary. Furthermore, it needs to be explained the effects of the risk assessment which is part of the MCDM as well. Since the FMEA assessment depends on alternative solutions and POVs (cf. Section 3.7.1) the FMEA only needs to be updated if there are changes regarding the individual alternative solutions. The necessity of them has been described in the previous paragraph concerning the linking between FM and SM.

In summary, the overall approach needs three essential steps which are written down prototypically in Algorithm 5.1. First, we need a FM to represent SPLs configurations. Subsequently, each feature of the individual SPLs configurations has to be linked with relevant system components of the underlying SM (cf. line 3) to identify correct alternative solutions. Finally, each of the features needs to be linked with POVs and associated goals (cf. line 4) to fulfil qualitative requirements. Afterwards, the MCDM can be performed for the individual SPL configuration.

Algorithm 5.1 Overall approach of the SPL based MCDM

```
1: procedure SPLOVERALLAPPROACH(splConfig, systemModel, sstModel)
2:     for all Feature f ∈ splConfig do
3:         setSystemComponents(f, systemComponents)
4:         setPOVs(f, POVs)
5:         setPathsPOVRoot(POVs)
6:     end for
7:     performMCDM(splConfig)
8: end procedure
```

5.3.2. Clustering of Semantically Equivalent Features

The preceding section covered how trade-offs of SPL configurations are calculated. However, complexity and functionality of software systems continue to increase (we talk about millions of SPL configurations) and thus the complexity needs to be reduced to calculate trade-offs of SPL configurations. Therefore, it is the aim of this section to extend the algorithm of Section 5.3.1 and to describe an algorithm to reduce complexity as far as possible in a way that all SST requirements are met and thus elements of the SSTM with same requirements are only included once for calculation of the optimal trade-off to reduce overhead. In general, the algorithm consists of five essential steps which are illustrated in Figure 5.4 and are essential part of this section. The green dotted relations refer to the linkings which have already been introduced in the preceding section whereas the blue relations refer to the algorithm of the current section.

Step 1: When selecting a SPL configuration only system components are considered which are used by the current SPL configuration due to the linking between FM and SM as described in the second step of the previous section. According to the SPL configuration the set of alternative solutions for determining the optimal

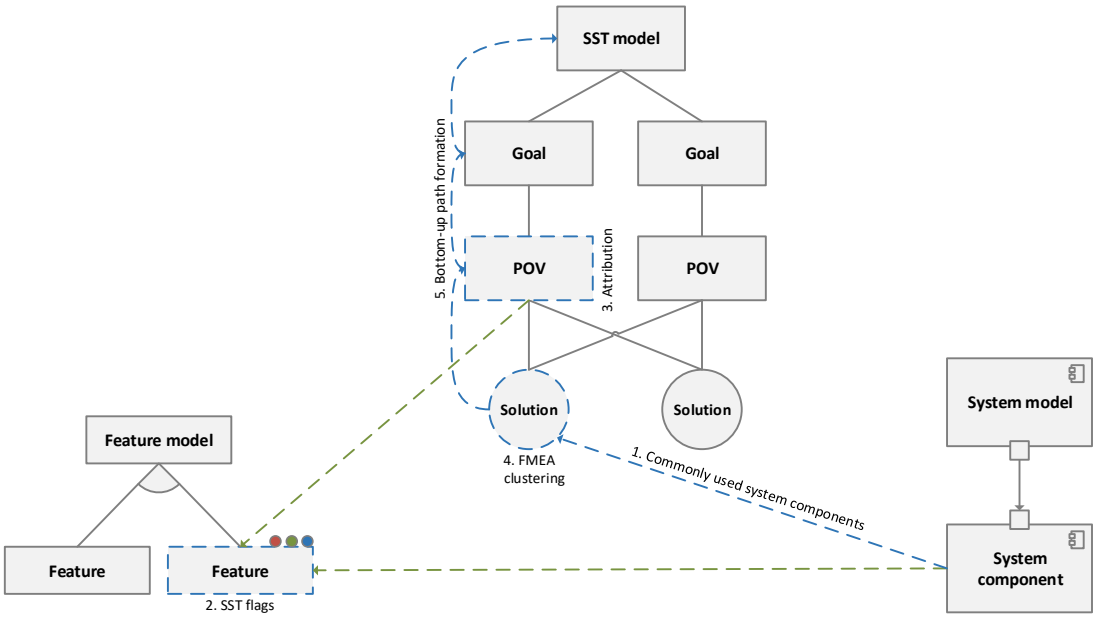


Figure 5.4.: Logical concept of clustering semantically equivalent features

trade-off may be extended to achieve better results for the MCDM in the end. For this purpose, we need to identify commonly used dependencies in the underlying SM. Afterwards, these dependencies have to be squared with the intuitive set of alternative solutions to take more of them into account and to use them for the MCDM. Let us assume there is a SPL configuration which uses *Feature* $F_{\#1}$ and *Feature* $F_{\#2}$, whereas *Feature* $F_{\#1}$ depends on *System component* $C_{\#1}$ and *Feature* $F_{\#2}$ depends on *System component* $C_{\#2}$. In this context, *Solution* $AS_{\#1}$ is derived from *System component* $C_{\#1}$ whereas *Solution* $AS_{\#2}$ depends on *System component* $C_{\#2}$. Besides, there is also *System component* $C_{\#3}$ which is associated with *System component* $C_{\#1}$ and *System component* $C_{\#2}$, i.e. *System component* $C_{\#3}$ is a commonly used system component. There is also a *Solution* $AS_{\#3}$ which is derived from *System component* $C_{\#3}$, i.e. we need to take *Solution* $AS_{\#3}$ into account for calculation of the optimal trade-off as well. The example which has been explained in this paragraph textually is illustrated in Figure 5.5.

Step 2: It is the aim of this algorithm to determine which features have similar SST demands to reduce elements for calculation of trade-offs. I.e., we need to determine the individual concerns (SST) for each feature of the corresponding SPL configurations by annotating them with relevant SST tags (cf. Figure 2.9). For instance, there may be features which are safety- and timing-critical or features which are only safety-critical. Since there is a linking between features and POVs it is ensured that each concern is covered by at least one POV. If there are two or more POVs with equivalent characteristics but different concerns for the features clustering is not possible since semantic characteristics are violated. The reason is that POVs are applied by individual features. If the individual POVs are used

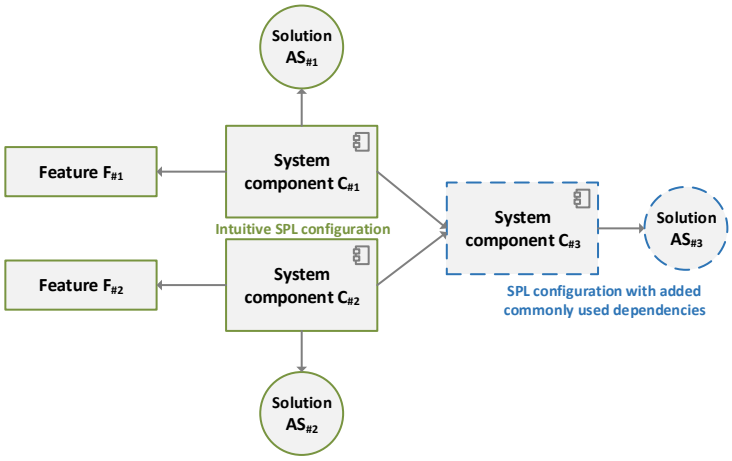


Figure 5.5.: Correlations between commonly used system components

by features with different SST requirements the POVs cannot be clustered for complexity reduction since the corresponding POVs are used in different ways.

Step 3: The last two steps covered necessary prerequisites for the actual complexity reduction. The third step is now responsible for semantic analysis of POVs. For this purpose we need a KPI based attribution as already proposed in Definition 4.3., i.e. there are one or more triples for each POV which consist of *attribute key*, *attribute function* and *attribute threshold*. Only when two or more POVs have identical attribution functions all further necessary steps for complexity reduction can be continued. For instance, if there are two POVs with attribution function *(timeLimit, max, 200)* respectively:

1. *The probability of detecting a threatening obstacle in time is acceptable for ACC video camera* and

2. *The probability of detecting a threatening obstacle in time is acceptable for LA video camera*.

Since in both cases the attribution function requires a time limit of 200 ms the prerequisites are fulfilled so far and the next steps can be continued to accomplish the complexity reduction.

Step 4: The last prerequisite before the complexity reduction is accomplished concerns risk assessment by means of the FMEA. As described in Section 3.7.1 the FMEA is performed for individual alternative solutions depending on corresponding POVs. To apply final complexity reduction it is necessary to check that risk assessment between individual alternative solution and POV has been classified with same *risk of error* (cf. Table 2.1). If the risk of error has been identically classified for all the alternative solutions of a POV and this condition applies to at least one more POV the complexity reduction can be performed if all preceding

requirements are fulfilled. Finally, there is one last question: Which POV is used for the calculation of the trade-off and which is eliminated? To achieve the best possible result for the MCDM the average RPN values of the FMEA are used and the best one is applied for the analysis whereas the remaining POVs will be eliminated. For instance, if there are two solutions and two POVs as listed in Table 5.2, $POV_{\#1}$ has an average RPN of 38 whereas $POV_{\#2}$ has an average RPN of 72,5. Furthermore, $POV_{\#1}$ and $POV_{\#2}$ have identical risk of error for both alternative solutions. Therefore, $POV_{\#1}$ is used for the MCDM and $POV_{\#2}$ is eliminated.

Alternative Solution	POVs	RPN	Risk of error
Solution $AS_{\#1}$	$POV_{\#1}$	36	Acceptable
	$POV_{\#2}$	70	Medium
Solution $AS_{\#2}$	$POV_{\#1}$	40	Acceptable
	$POV_{\#2}$	75	Medium

Table 5.2.: FMEA clustering of semantically similar features

Step 5: Before the trade-off for the individual SPL configuration can be calculated the corresponding paths within the underlying SSTM needs to be enabled. For this purpose, all remaining POVs have to be enabled. Subsequently, all these POVs enable all overlying goals up to the root. Afterwards, all remaining alternative solutions (cf. step 1) have to be linked with the enabled POVs (cf. Section 3.8) unless it is already done. Finally, the trade-offs for individual SPLs configurations can be determined.

The five steps which have been described in the last paragraphs are summarised in Algorithm 5.2. Thereby, step 1 is realised in line 2 when the method *enableSolutions* is called to determine commonly used system components and thus to enable corresponding solutions. Line 4 and 5 cover step 2 and are responsible for assigning SST concerns to the individual features. Subsequently, the verification regarding semantics is realised in line 7 and 8. Step 4, i.e. the FMEA clustering is checked in line 13 before the final paths will be built in line 16 and the trade-off is calculated.

Definition 5.2 (Complexity Reduction of MC in SPLs)**.** The level of complexity reduction when using SPLs for MC to calculate trade-offs is defined as:
$1 - \frac{\sum_{i=1}^{l} m - \sum_{j=1}^{n} v}{w}$ with following interpretation of the variables:

- l: Number of paths after complexity reduction.
- m: Number of goals and POVs in the corresponding paths.
- n: Number of commonly used goals within different paths.
- v: Indicates the number of multiple use of commonly used goals.
- w: Number of goals and POVs before complexity reduction.

Algorithm 5.2 Clustering approach of the SPL based MCDM

```
1:  procedure SPLCLUSTERINGAPPROACH(splConfig, systemModel, sstModel)
2:      enableSolutions(splConfig, systemModel)
3:      for all Feature f ∈ splConfig do
4:          setSSTFlag(f)
5:          checkSSTFlag(f)
6:          for all POV pov ∈ getLinkedPOVs(f) do
7:              setAttribution(pov)
8:              checkAttribution(pov)
9:          end for
10:     end for
11:     for all Solution s ∈ getEnabledSolutions(sstModel) do
12:         for all POV pov ∈ getLinkedPOVs do
13:             checkFMEAClustering(s, pov)
14:         end for
15:     end for
16:     disablePOVsAndPathFormation(splConfig, sstModel)
17:     performMCDM(splConfig)
18: end procedure
```

5.4. Extension of Change Impact Ruling

So far, change impacts which only concerns calculating a standalone MCDM have been considered. It is the aim of this section to take change impact ruling into account which enables MCDMs of SPL configurations. Therefore, we introduce four new rule types: *FM* → *FM*, *SM* → *FM*, *SSTM* → *FM* and *FM* → *SSTM*.

5.4.1. FM to FM

The FM is a hierarchically structured model and is an essential part of the MCDM when realising MC in SPL. If functional requirements and thus the FM is changed by external influences, e.g. due to market demands it may have effects on model elements within the FM itself. When applying rules with the syntactical scheme `A.X` → `B.Y` (cf. Section 4.3) which are listed in Table 5.3 it is distinguished between three rule types [LSB15]: *Feature* → *Feature* (↓), *Feature* → *Feature* (↑) and *Feature* → *Feature* (require/exclude). I.e., there are impact rules which calculate effects bottom-up and top-down whereas bottom-up rules are more lightweight than top-down rules. Furthermore, there are rules which take *require* and *exclude* relationships between different hierarchy levels into account.

When extending a feature there are no effects on child features or parent features in BC if there is only a marginal extension. However, in WC the child features and parent features need to be modified since the extension may have any functionality of them and thus it is mandatory to adapt them. In another case, if a lightweight

Source/Target element	BC	WC
Feature → Feature (↓)	A.ext → B.noChange	A.ext → B.mod
	A.mod → B.noChange	A.mod → B.mod
	A.del → B.del	A.del → B.del
Feature → Feature (↑)	A.ext → B.noChange	A.ext → B.mod
	A.mod → B.noChange	A.mod → B.mod
	A.del → B.mod	A.del → B.ext
Feature → Feature (requ./excl.)	A.ext → B.noChange	A.ext → B.ext
	A.mod → B.noChange	A.mod → B.mod
	A.del → B.noChange	A.del → B.del

Table 5.3.: Impact rules of dependencies FM → FM

feature is modified the corresponding child features and parent features remain unchanged. The situation is different if a mighty feature is modified. In this case, the corresponding target features need to be modified since missing information of the source features has to be transferred to the target features. When deleting a feature the underlying child features are deleted as well since they cannot exist without parent features. However, if a feature is deleted the corresponding parent features are modified in BC and extended in WC. This is justified by the fact that modifying a feature is associated with less effort than to develop a extending feature. FMs support the functionality of *requiring* or *excluding* features which are located on different hierarchy levels. To extend a feature, which has a *require* or *extend* connection, has no effects on target feature in BC. The target feature needs to be extended if the source feature is more important since the information of extension of the source feature has to be transferred to the target feature. When modifying a source feature there are no impacts for lightweight source features. However, target features need to be modified in WC since information get lost otherwise. The last case covers deleting a feature which is connected via *require* or *exclude* relation. In this case, the target feature remains unchanged in optimal case. Only in WC the target feature is deleted since dependencies are too strong.

To avoid potential conflicts when applying impacts rules, which are listed in Table 5.3, it is mandatory to define corresponding preferences for BC and WC:

$$\texttt{mod} \succ_{BC} \texttt{ext} \succ_{BC} \texttt{del} \text{ and}$$
$$\texttt{ext} \succ_{WC} \texttt{mod} \succ_{WC} \texttt{del}$$

In both cases deleting has lowest priority since information get lost. In optimal case it is better to modify features than to extend it since it is more difficult to provide new features than to adapt existing ones. Only in WC it is higher prioritised to extend features than to modify it, since the WC analysis is not intended to conserve resources.

All the change impact rules which have been defined within this section have to be applied by means of a individual algorithm. This algorithm is specified

Algorithm 5.3 Calculation of impacts of dependencies FM → FM

```
1:  procedure CALCULATIONIMPACTS-FM-FM(feature, operation)
2:      for all f ∈ getChildFeatures(feature) do
3:          if checkPrefAndApplyRule(feature, f) then
4:              CalculationImpactsFM-FM(f, getOperation(f))
5:          end if
6:          if checkReqExcl(feature, f) then
7:              CalculationImpactsFM-FM(f, getOperation(f))
8:          end if
9:      end for
10:     Feature f ← getParentFeature(feature)
11:     if checkPrefAndApplyRule(feature, f) then
12:         CalculationImpactsFM-FM(f, getOperation(f))
13:     end if
14:     if checkReqExcl(feature, f) then
15:         CalculationImpactsFM-FM(f, getOperation(f))
16:     end if
17: end procedure
```

in Algorithm 5.3 and mainly consists of methodology which defines top-down and bottom-up rules and cross-links between individual features on different levels. First, all child features of an individual feature are determined and the preferences, which have been explained in the last paragraph, are checked to avoid conflicts. Subsequently, the corresponding impact rules are applied top-down and the method is invoked recursively with the new operation or effect type from the current target feature (cf. lines 4-6). In addition, it is revised whether there are *require* or *exclude* relations and the corresponding impact rules are applied if applicable. Afterwards, the method is invoked recursively with the corresponding target feature and effect type (cf. lines 7-9). Following, the parent feature is determined based on the current feature and the same procedure is accomplished once again bottom-up for the relation current feature → parent feature (cf. lines 11-17). This algorithm has the same structure regarding method invokes, i.e. the complexity is $\mathcal{O}(|V| + |E|)$ for all relevant features $|V|$ and connections $|E|$ between them.

5.4.2. SM to FM

As already described in Section 5.3 there is a dependency between SM and FM or more precisely the FM depends on the underlying SM to derive commonly used system components. The corresponding impact with the syntactical scheme `A.X` → `B.Y` are listed in Table 5.4 [LSB15]. In this context there are only impact rules of the type: *Component* → *Feature*, i.e. in this case there are no hierarchy related rules.

When extending a system component, e.g. by adding new ports, to consider

Source/Target element	BC	WC
Component → Feature	`A.ext → B.mod`	`A.ext → B.ext`
	`A.mod → B.noChange`	`A.mod → B.mod`
	`A.del → B.noChange`	`A.del → B.del`

Table 5.4.: Impact rules of dependencies SM → FM

sub-components, the corresponding features which are linked with the system components need to be modified in BC or extended in WC. There is less effort to modify a feature than to extend it since extending yields providing new functionality whereas modifying means to update existing ones with minor changes. In other cases if the source components are modified the corresponding target features only need to be modified in case of major changes. For instance, the use of a new sensor technology requires adapting target features to this technology. Otherwise the individual target features are not adapted. Deleting a source system component yields also deleting target feature if we consider a maximal set of effects. For instance, if a feature depends only on one system component the corresponding feature needs to be deleted. Otherwise there are no notable changes. As already mentioned in the preceding section, conflicts can arise when applying several change impact rules. Therefore, it is necessary to define preferences to get correct results:

$$\texttt{mod} \succ_{BC} \texttt{ext} \succ_{BC} \texttt{del} \text{ and}$$
$$\texttt{ext} \succ_{WC} \texttt{mod} \succ_{WC} \texttt{del}$$

Deleting a source system component has lowest priority since it means that the knowledge has to be replaced in other ways. In BC it is easier to modify a system component than to extend it. To modify a system component yields in many cases changing the manufacturer where the technology is identical. However, in the less common cases it is better to provide new system components than to update an existing technology if the set of effects should be as minimal as possible.

Algorithm 5.4 Calculation of impacts of dependencies SM → FM

```
1: procedure CALCULATIONIMPACTSSM-FM(component, operation)
2:     for all feature ∈ getAssociatedFeatures(component) do
3:         if checkPref(component, feature) then
4:             applyRule(component, feature)
5:         end if
6:     end for
7: end procedure
```

This section covers impacts for dependencies between two different model types, i.e. we first need to fetch all features which have been linked to the component. The calculation method *getAssociatedFeatures* is invoked by means of that component. Subsequently, it is mandatory to synchronise with the above specified preferences. Only if the method *checkPref* returns `true` the corresponding impact rule can be

applied. Finally, the complexity of Algorithm 5.4 is analysed: The algorithm is not called recursively, i.e. there is a linear complexity of $\mathcal{O}(|V| + |E|)$ for all relevant system components and features $|V|$ and $|E|$ connections between them.

5.4.3. SSTM to FM

As already stated in Section 5.3 there is a linking between SSTM and FM. This dependency is necessary to enable calculation process of MC in SPLs. When changing a POV due to external influences it will consequently have effects on the corresponding linked features, i.e. there is no hierarchy between the linking. The necessary impact rules are listed in Table 5.5 and cover change impacts of the type *POV* $\rightarrow$ *Feature*.

Source/Target element	BC	WC
POV $\rightarrow$ Feature	`A.ext` $\rightarrow$ `B.mod`	`A.ext` $\rightarrow$ `B.mod`
	`A.mod` $\rightarrow$ `B.noChange`	`A.mod` $\rightarrow$ `B.mod`
	`A.del` $\rightarrow$ `B.noChange`	`A.del` $\rightarrow$ `B.del`

Table 5.5.: Impact rules of dependencies SSTM $\rightarrow$ FM

When extending a POV it means that the current POV is converted into a goal, i.e. the corresponding target feature needs to be modified since there is no linking between goal and feature. In this context, the new POV which has been extended or another POV needs to be set once more. In other cases, if individual lightweight POVs are modified there are no effects on the corresponding features which belong to SPL configurations. Only if more important POVs are modified the corresponding features need to be adapted. For instance, if there is a POV *Road sign recognition is acceptably safe* which is linked with the feature *Road sign recognition* and the aforementioned POV is modified to *Speed limit road sign recognition is acceptably safe*, the feature *Road sign recognition* needs to be modified. In this case, the feature is either renamed to *Speed limit road sign recognition* or the feature needs a linking with additional POVs for recognition of danger road signs. The last change impact rule covers deleting a POV. In this case, the corresponding target feature is only deleted if the feature only depends on the source POV. Applying different change impact rules may yield conflicts, therefore we need to define preferences in order to avoid those conflicts over several iterations (e.g. *FM* $\rightarrow$ *FM* and *SSTM* $\rightarrow$ *FM*):

$$\texttt{mod} \succ_{BC} \texttt{ext} \succ_{BC} \texttt{del} \text{ and}$$
$$\texttt{mod} \succ_{WC} \texttt{ext} \succ_{WC} \texttt{del}$$

In this application there is no difference between BC and WC. Thereby, deleting has the lowest priority since deleting a feature may yield destroying a SPL configuration. Furthermore, modifying is ranked better than extending since extending a feature may require adapting the corresponding SPL configuration. This does not apply for modifying a feature.

Algorithm 5.5 Calculation of impacts of dependencies SSTM → FM

```
1: procedure CALCULATIONIMPACTSSSTM-FM(component, operation)
2:     for all feature ∈ getAssociatedFeatures(pov) do
3:         if checkPref(pov, feature) then
4:             applyRule(pov, feature)
5:         end if
6:     end for
7: end procedure
```

Similar to the preceding section dependencies between two different model types need to be considered, namely the FM and the SSTM. Algorithm 5.5 specifies the methodology of the corresponding change impact ruling. First, we need to determine all the features which have been associated with the corresponding POVs. The method body *getAssociatedFeature* (cf. line 2) is responsible for determining all involved features depending on relevant POVs. Subsequently, it is mandatory to check preferences which have been defined within this section. If positive, the corresponding change impact rule which is part of Table 5.5 can be applied. The complexity of this algorithm is linear, i.e. $\mathcal{O}(|V| + |E|)$ steps are necessary to perform this algorithm where $|V|$ corresponds to all relevant POVs and features with $|E|$ connections between them.

5.4.4. FM to SSTM

Changes to the FM yield fundamental effects on the SSTM and thus on the calculation of trade-offs. The last sections covered defining impact ruling sets to easily calculate effects with minimal effort. In this case it is not possible to define a rule set since calculation of trade-offs in SPLs in consideration of MC depends on several steps and factors (cf. Section 5.3.2). When changing elements of the FM there are three opportunities:

1. *SST flags of the features*: by modifications of qualitative requirements SST flags of the individual features may change. This consequently has impacts on the subsequent semantic clustering.

2. *Adapting POV links of a feature*: POVs may change due to external influences (functional requirements) and have to be adapted accordingly. This entails adjusting the links between individual features and POVs.

3. *Extending a feature*: by extending a feature, functionalities of existing features may be redistributed to new features. This may affect both the SST flags and the modification of POV links.

If one of the aforementioned changes is made, a new calculation has to be performed as described in Section 5.3.2 to ensure the quality of the MCDM. In this

case, there are too many interdependencies concerning the clustering of semantically equivalent features which have been described in Section 5.3.2 and would falsify the final result of the MCDM.

5.5. Example

The current section exemplifies the clustering of semantically equivalent features for the calculation of trade-offs. The example is illustrated in Figure 5.6 and represents the necessary models for calculating trade-offs regarding ADASs.

First, there is a FM which allows the selection of three ADASs: ACC, LA and LDP. In this case study, we choose the SPL configuration with the feature *ACC*. Furthermore, there is an underlying SSTM consisting of necessary, goals, POVs and alternative solutions. Moreover, a SM is used to calculate commonly used system components (cf. step 1 of Section 5.3.2) and thus to enable corresponding alternative solutions which are essential for calculating trade-offs. When checking the ACC dependencies one realises that the ACC system component is using a *radar sensor* and a *wide-angle camera*. However, the *wide-angle camera* is used by the LA and the LDP. Since the LDP uses an additional *ultrasonic sensor*, only the ACC and LA system component is used to enable the *ACC* and *ACC + LA* alternative solutions for calculating trade-offs. All three ADASs are SST-critical, i.e. they are tagged with corresponding SST flags (cf. step 2 of Section 5.3.2). I.e., all of them provide the same combination of SST concerns. Subsequently, attribution functions need to be assigned to the POVs to ensure semantic correctness (cf. step 3 of Section 5.3.2). These KPIs can be taken from Figure 5.6. The result is that $POV_{\#1} \equiv POV_{\#3}$ as well as $POV_{\#2} \equiv POV_{\#4}$. The reason is that $POV_{\#1}$ and $POV_{\#3}$ require a SIL level of at least 3 and $POV_{\#2}$ and $POV_{\#4}$ require an encryption algorithm with a key length of at least 128 bit. Since the linked features *Adaptive Cruise Control* and *Lane Assist* provide same SST concerns semantic equivalence between $POV_{\#1}$ and $POV_{\#3}$ as well as between $POV_{\#2}$ and $POV_{\#4}$ is almost fulfilled. A FMEA clustering is still missing (cf. step 4 of Section 5.3.2). The corresponding RPN values, average values, risks of error are listed in Table 5.6.

POV	**RPN *ACC***	**RPN *ACC + LA***	**∅ RPN**	**Risk of error**
$POV_{\#1}$	36	40	38	Acceptable
$POV_{\#2}$	60	64	62	Medium
$POV_{\#3}$	42	40	41	Acceptable
$POV_{\#4}$	70	75	72,5	Medium

Table 5.6.: MC in SPLs: FMEA values of the example

$POV_{\#1}$ and $POV_{\#3}$ as well as $POV_{\#2}$ and $POV_{\#4}$ have identical risk of error, i.e. the two pairs are semantically equivalent. Since the average RPN of $POV_{\#1}$ < RPN of $POV_{\#3}$ and the average RPN of $POV_{\#2}$ < RPN of $POV_{\#4}$, $POV_{\#1}$ and $POV_{\#2}$ are enabled for calculating trade-offs. Subsequently, all goals bottom-up from the

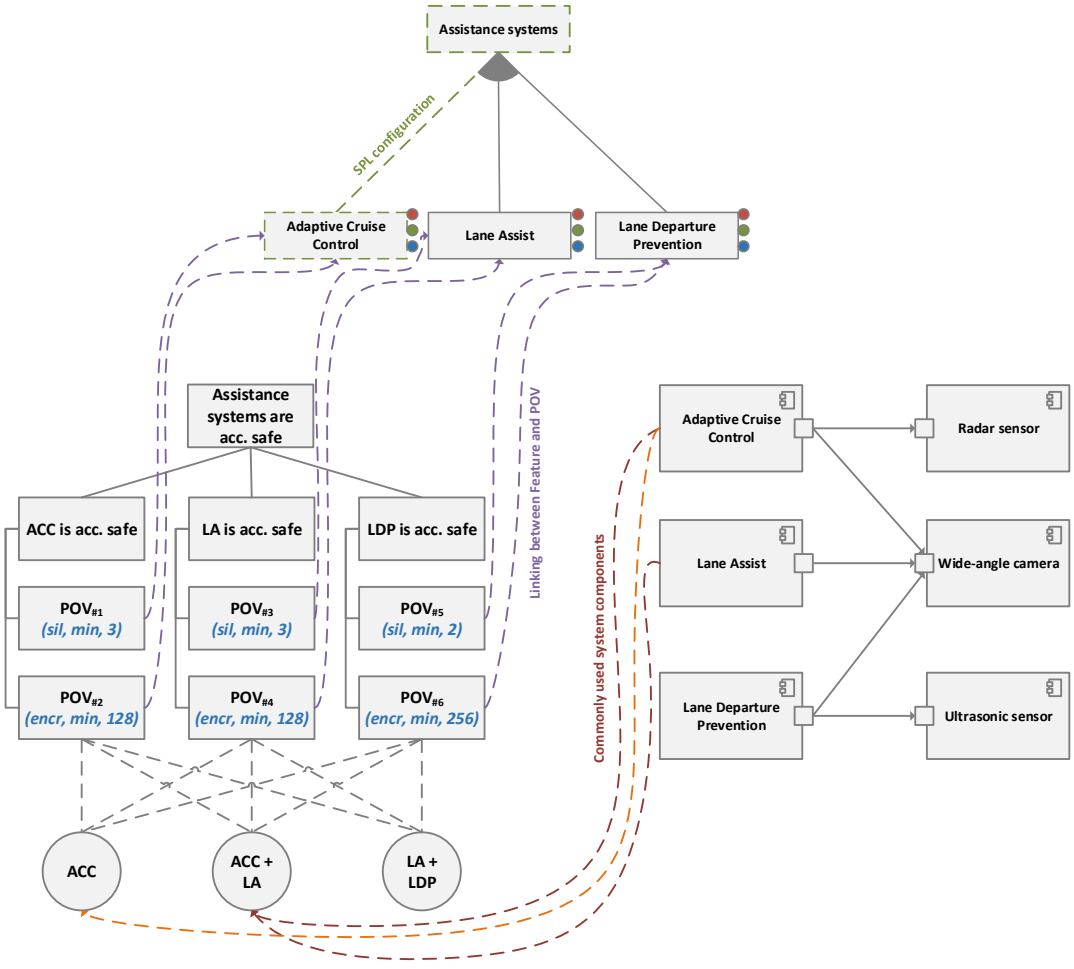


Figure 5.6.: Example of using MC in SPLs

two POVs are enabled up to the root. Consequently, a suitable trade-off can be calculated using the enabled branches within the SSTM.

Finally, the degree of complexity reduction is determined by applying Definition 5.2. In this way, we get a complexity reduction of $1 - \frac{3+3-1-1}{10} = 0{,}6$ The complexity has been reduced by more than 50% in compliance with qualitative requirements. The entire SSTM contains 10 goals and POVs whereas the selected SPL configuration only needs 4 nodes.

5.6. Related Work

There are some scientific publications regarding MC in SPLs which are presented within this section. First, there are publications which cover SPLs in general and are differentiated to the work of this thesis. Furthermore, related work is presented which covers consideration of MC in SPLs. Finally, we discuss publications which handle complexity reduction of SPL in different contexts.

SPLs in General

The work of [PBD05] serves as baseline for SPLs. In this thesis, definitions and scopes of SPLs are introduced in detail. This chapter of the thesis extends the work of [PBD05] by safety-critical concerns, e.g. SST. Furthermore, there is no clustering of semantically equivalent elements in [PBD05]. In [Poh+18], the extent of FMs is measured and analysed by means of worst-case execution analyses to improve state-of-the-art analysis tools. The approach improves complexity in the context of state-of-the-art analysis tools whereas this chapter of the thesis is focused on complexity reduction in safety-critical environment. Furthermore, our approach is focused on the clustering semantically similar features.

Consideration of MC in SPLs

The work of [Bra+12] aims to optimise model-driven SPL engineering process to certify embedded systems in safety-critical environment. For this purpose, a meta-model is proposed to enable the certification process. The work of this thesis is not only focused on certification of safety-based systems. We also consider influences and interrelations between individual safety-critical systems. Furthermore, we reduce complexity of SPLs while maintaining all SST requirements. The paper of [LDL07] proposes an approach which integrates a model-based safety analysis for SPLs. The authors use state-based models to realise SPLs and to integrate model-based safety analysis within them. In this thesis we additionally provide an approach which provides clustering of semantically equivalent POVs to calculate trade-offs taken SST into account. Finally, there is another publication of [Met+07], dealing with MC in SPLs. Thereby, it is the aim to "analyze whether the product line artifacts are flexible enough to build all the systems that should belong to

the product line." [Met+07] In this context, the product line artefacts correspond to the individual concerns. This work does not primarily consider safety-critical concerns. In contrast, it is not aimed to calculate trade-offs on the basis of SPLs.

Complexity Reduction of SPLs

[Li+18] propose an approach which analyses relations between variants, i.e. features within a FM to finally reduce the number of them. In this thesis, the complexity is reduced by linking the FM with a underlying SM and SSTM. In this way, we use an equivalence-based approach to reduce complexity. Moreover, we considered safety-critical requirements, e.g. SST whereas Li et al. [Li+18] only considered economical requirements. The authors of [Pol+12] presented an approach how to realise model-based SPL techniques into a framework which can handle with a large number of SPLs. In this context, complexity and dependencies between individual requirement, test and implementation artefacts are analysed to derive products therefrom. In this thesis the complexity is reduced by means of a equivalence-based clustering algorithm which is not part of [Pol+12]. Moreover, their approach has not been extended to calculate trade-offs taken SST into account. Furthermore, there is the work of [HK18] which proposes a concept to optimise feature selection to choose the elements which are reused most commonly. The focus of [HK18] is on economic goals such as cost minimisation. In contrast, in this chapter of the thesis safety-critical requirements are preferred. In this context, it may have fatal consequences if only most commonly reused features are considered. Finally, there is the paper of [Ben+08] with the aim to "[...] propose an approach to support dynamic or runtime variability in systems that must adapt dynamically to changing runtime context. The approach is founded on reflective component-based technologies to support the dynamic variability at the architecture level." [Ben+08] In this paper the focus is on dynamic runtime variability whereas this chapter of the thesis is focused on complexity reduction of SPLs taken MC into account by clustering semantically equivalent features.

Part III.

Evaluation and Conclusion

6 Realisation and Evaluation

The current chapter provides the reader with details on implementation and evaluation. First, Section 6.1 gives an introduction on the Eclipse frameworks, Eclipse Modeling Framework (EMF) and Sirius and how they interact. Subsequently (Section 6.2), details about implementation are clarified. This information is provided by means of underlying prototypical meta-models. Section 6.3 covers the actual evaluation which is performed by means of selected scenarios to show satisfiability of individual quality attributes. Finally, two selected case studies are presented in Section 6.4 to demonstrate the approaches of this thesis by means of case studies.

6.1. Eclipse EMF and Sirius

The concepts, which have been presented in the preceding three chapters, have been implemented prototypically as a proof of concept. The EMF and Sirius framework has been used to realise the presented concepts. Figure 6.1 illustrates the three tiers of EMF and Sirius.

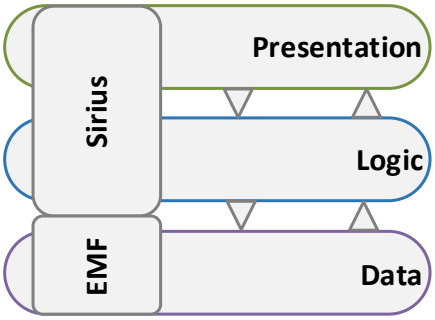


Figure 6.1.: Three-tier architecture of EMF and Sirius [Fen16]

The three-tier system architecture consists of *data*, *logic* and *presentation*. The data tier is covered by EMF whereas logic and presentation is covered by Sirius. It is built upon a model which is created by means of the EMF meta-model. Sirius provides the diagram editor which represents the presentation tier. On the basis of the individual EMF meta-models of the concepts Sirius creates the so called business model which is responsible for representing the logic tier. Sirius provides a representation model which is used to outline the business model in a

graphical editor. This part covers Sirius' presentation tier. [EF19] Sirius does not strictly comply with multi-tier architecture since there is direct access to EMF data structures from the Sirius presentation tier.

6.2. Prototypical Implementation

The concept of this thesis is divided into three parts: MCDM, change impact analysis and MC in SPLs. To realise each of them, an underlying meta-model was developed. Hereinafter, the individual meta-models are explained in more detail.

6.2.1. Multi-Concerns and Multi-Criteria Decision Making

As already described in Section 3.4.1, the SSTM is an essential part of the MCDM, i.e. a trade-off cannot be calculated without any SSTM. When recalling Chapter 3 the following information is required to realise a MCDM:

1. There is an hierarchically structured SGH which enables modelling of SST goals.
2. For each goal there is a matrix which defines importance of the underlying sub-goals or POVs.
3. There is a FMEA risk assessment for each POV depending on the individual alternative solutions.
4. Each security goal can be optionally extended by an ADT to enable an ADTA.

All the information listed above has to be transferred into an EMF meta-model which is illustrated in Figure 6.2. First, there is class *SSTModel* which represent the root element, i.e. the diagram itself. A unique name has to be assigned to the element. All model nodes, i.e. goals, POVs and alternative solutions are represented by the class *SSTObject*, characterised by an identifier. Furthermore, there are two classes: *SSTObjectCanSupport* and *SSTObjectCanBeSupported*. Objects of type *SSTObjectCanSupport* can be refined by objects of type *SSTObjectCanBeSupported*. Hence, the class *SSTObjectCanBeSupported* is complemented by *ratioMatrixData* which represents the AHP pairwise comparisons. Moreover, there are two further classes *SSTGoal* and *SSTSolution* where the first one represents goals and POVs. The *SSTGoal* needs information about the corresponding concern and the extended ADT diagram if applicable. Moreover, the boolean variable *enabled* indicates whether the goal or POV should be taken into account for the calculation of trade-offs. Since a goal or POV *can support* and *can be supported* it is derived from the corresponding two presented classes. The class *SSTSolution* has two attributes: The parameter *solutionDataContainer* covers risk assessment by means of the FMEA as described in Section 3.7.1. Furthermore, it is considered whether the individual solutions should be taken into account for the MCDM. Since an alternative solution only provides *can support* functionality it is derived from

SSTObjectCanSupport. Alternative solutions depend on system components, i.e. an association between *SMComponent* and *SSTSolution* where *SMComponent* refers to class of an external meta-model which represents system modelling

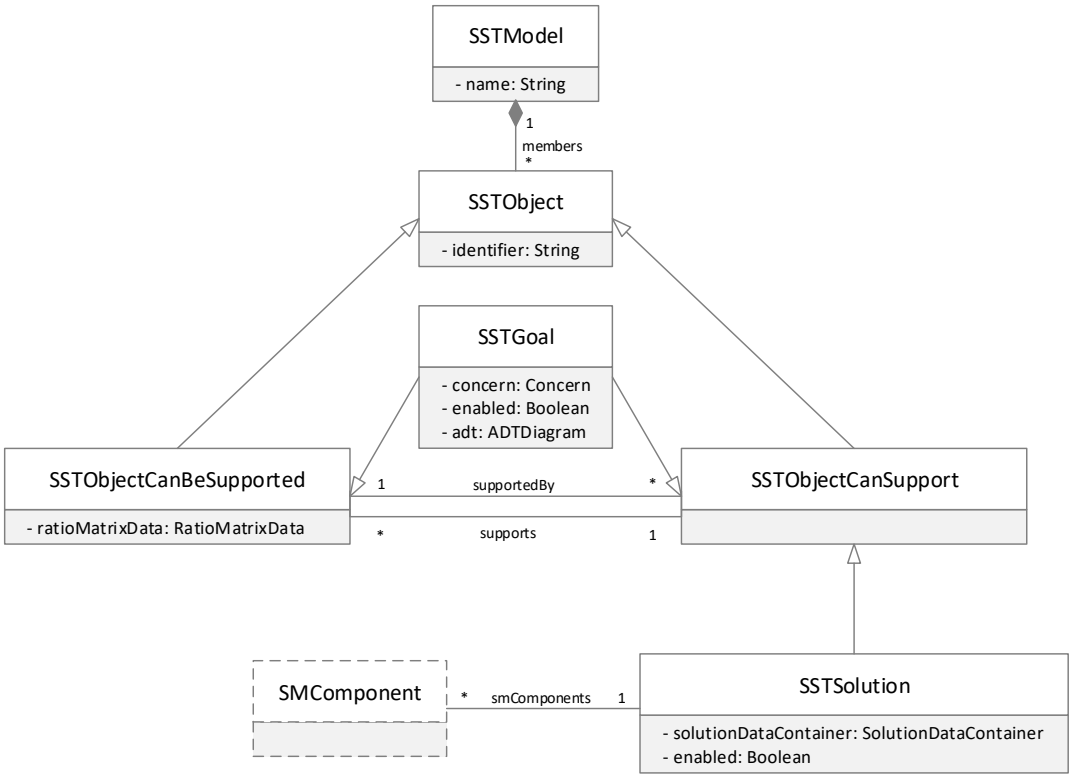


Figure 6.2.: Abstract meta-model of the SSTM

As already described in the preceding paragraph, a *SSTGoal* can be optionally extended by an ADT diagram to perform an ADTA and thus to determine suitable CMs. When recalling Section 3.4.2 there are three essential constituents which need to be considered:

1. *Attack node*: Represents an external attack.
2. *Action node*: Specifies how an attack is realised.
3. *CM node*: Developed method to avoid actions.

The EMF meta-model is illustrated in Figure 6.3. First, there is a class called *ADTDiagram* which represents the root element. There is an attribute *name* which represents an unique name for *ADTDiagram*. The class *ADTDiagram* is associated with an abstract class *Node* which only identifies attack, action and CM nodes, i.e. the classes *AttackNode*, *ActionNode* and *CMNode* are derived from class *Node*. The class *AttackNode* has an attribute *logicalOperator* which identifies the logical operation (*AND, OR*) of the attack node. As described in Section 3.7.2 some

parameters are needed to calculate ROI and ROA in context of ADTA. Therefore, the variable *af* is assigned to class *AttackNode*. For the same reason, parameters *ef* and *aro* are assigned to class *ActionNode* whereas variables *csi, rm, cost, loss* are assigned to class *CMNode*. Finally, the relations between the individual node types are defined by associations. An attack node is represented by any number of action nodes whereas an action node is counteracted by any number of CM nodes.

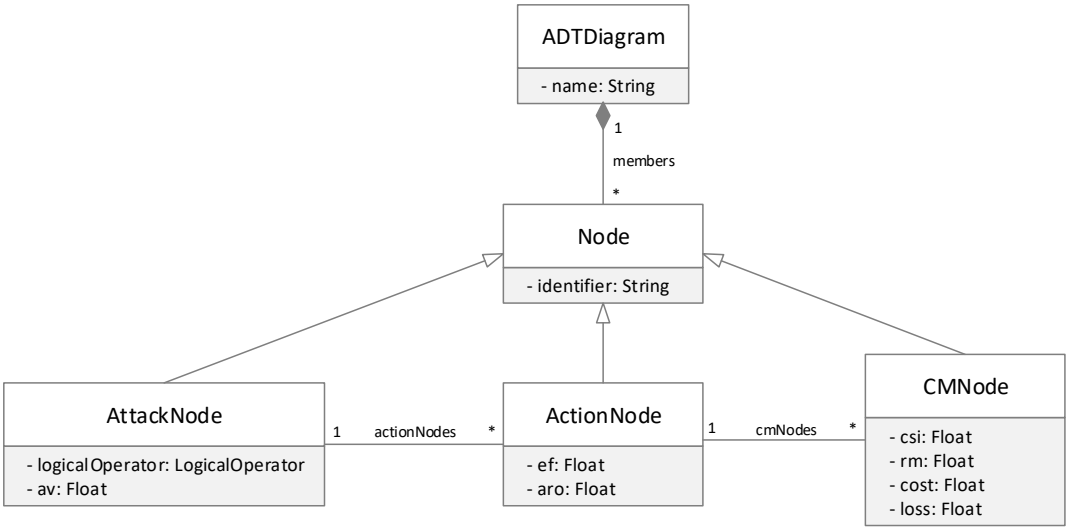


Figure 6.3.: Abstract meta-model of the ADT

6.2.2. Change Impact Analysis

The change impact analysis as proposed in Chapter 4 concerns different model types including SSTM, ADT, SM and FM. However, each model has its own characteristics. Nevertheless, we define an universal EMF meta-model which enables calculation of change impacts independent of the respective model type. In advance we want to collect all necessary information which should be covered by the meta-model:

1. There is a stakeholder which initiates a change request.

2. A change impact can be calculated by two algorithms: A structural approach or a KPI based approach.

The EMF meta-model, which provides the necessary information, is illustrated in Figure 6.4. First, there is a class *ChIAModel* which represents the root element with an unique name. Within this *ChIAModel* we need any number of stakeholders which are identified by unique names. Furthermore, a class *ChangeRequest* is created which has a large number of attributes to enable calculation of change impacts. At first, there is a textual description of the change request to assign it to

stakeholders. Moreover, there is a variable methodology which determines the type of algorithm, i.e. a calculation of impacts is performed on a structural or KPI based approach. As described in Section 4.5.2 for calculation of KPI based impacts we need the attributes *deltaFMEA* and *allAttributes*. In case of structural impacts we need a trigger operation, i.e. an operation (`ext`, `mod` or `del`) by which the change impact analysis is started. The impact rules itself are implemented by means of Java code. For both methodologies there has to be an affected model element which concerns a model node of the SSTM, ADT, SM or FM.

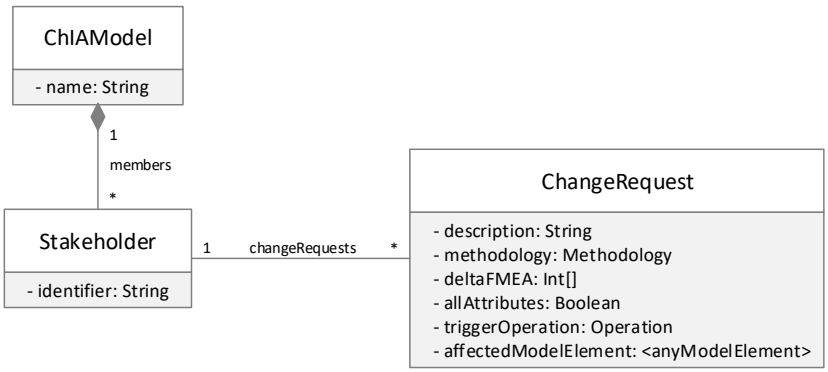


Figure 6.4.: Abstract meta-model of the ChIA

6.2.3. Multi-Concerns in Software Product Lines

To enable MC and thus MCDM for SPLs it is essential to define FMs. I.e., we have to define an EMF meta-model which has the characteristics of a FM to enable MC in SPLs. When recapitulating Chapter 5 the meta-model has to fulfil the following requirements:

1. Features are annotated with SST flags to determine whether a feature is safety-critical or not.

2. There is a tag for (sub-)features whether they are mandatory. Moreover it has to be possible to define logical operator for them.

3. Features can require or exclude another features.

4. For the complexity reduction features are linked with system components and POVs.

Figure 6.5 represents the EMF meta-model which fulfils all listed requirements. First, there is a class called *FMDiagram* which represents the root element. This *FMDiagram* has an unique name. All child nodes of *FMDiagram* are derived from class *FMObject* which have to be identified uniquely. Further, it is distinguished

between features, which can be parent features and sub-features. This context is realised by two classes: *FMFeatureCanBeParentFeature* and *FMFeatureCanBeSubFeature*. The first one is refined by *FMFeatureCanBeSubFeature*. Sub-features are extended by boolean variables. It is indicated whether a feature is mandatory or optional. Furthermore, logical combinations are specified for sub-features (*OR* and *XOR*). The main class *FMFeature* contains three boolean attributes for SST flags. Furthermore, there is a boolean attribute which indicates whether the corresponding feature is abstract or concrete. Moreover, there are several associations based on class *FMFeature*. On the one hand there are associations between *FMFeature* and *FMFeature* which represent *require* and *exclude* relations. On the other hand there are associations between *FMFeature* and *SMComponent* as well as *FMFeature* and *SSTGoal*. These associations are necessary for the complexity reduction.

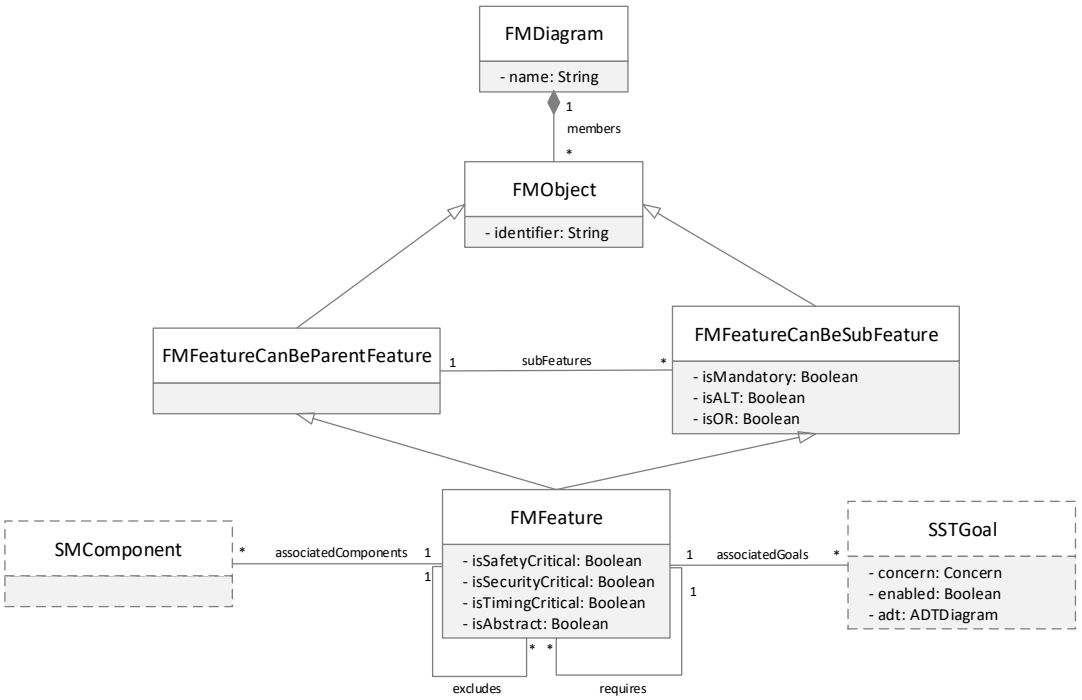


Figure 6.5.: Abstract meta-model of SPL modelling

6.3. Scenario Based Evaluation

This thesis is evaluated qualitatively, i.e. it is investigated by means of selected scenarios whether quality demands are fulfilled. In this context, the approach of this thesis as well as the SuD, which is analysed by the approach, is evaluated (cf. Figure 6.6).

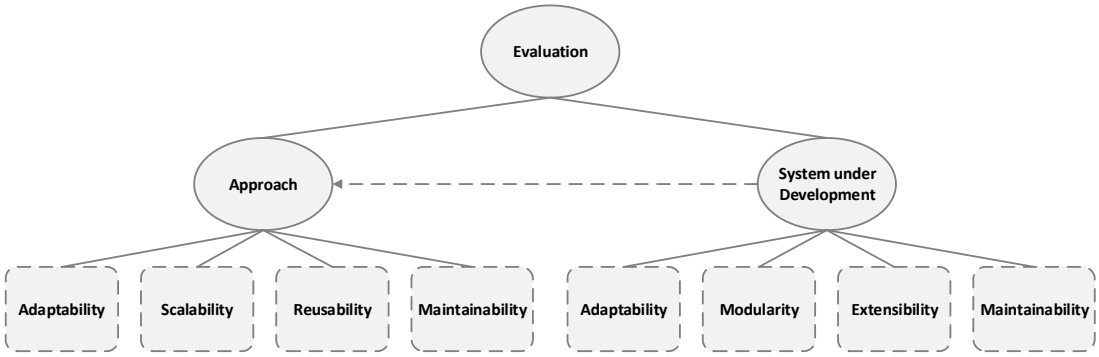


Figure 6.6.: Evaluation of the thesis

The evaluation investigates the following quality attributes:

1. *Adaptability:* The approach and SuD has to be modifiable to perform changes with minimal effort.

2. *Scalability:* The approach of this thesis has to be scalable.

3. *Reusability:* Individual modules within a system should be reusable to save overhead and resources.

4. *Maintainability:* In principle, it cannot be ruled out that a system is flawless. Furthermore, new features and change request should be considered. Therefore, maintainability has to be ensured.

5. *Modularity:* Various sub-modules are combined to form an entire module or system.

6. *Extensibility:* The SuD has to be extended or exchanged by further functionality at any time.

Each of these quality attributes is either evaluated by means of the approach of this thesis or the SuD which uses the approach of the thesis in the development phase. For this purpose, suitable scenarios are defined to determine corresponding quality demands. First, the evaluation is performed for the concept of this thesis. Subsequently, it is assessed in context of genuine systems. In both cases, the definition of the corresponding quality attributes is given first if it has not yet been defined. Following, the scenarios by which the quality attribute are evaluated is presented before the actual evaluation is performed.

6.3.1. Approach Evaluation

The current section covers evaluation of the developed concepts. In this context, quality attributes *adaptability*, *scalability*, *reusability* and *maintainability* are evaluated.

6.3.1.1. Adaptability

First, it is evaluated by means of selected scenarios whether the concept of this thesis fulfils the property of adaptability. Therefore, we first define the term *adaptability*:

Definition 6.1 (Adaptability)**.** According to [SC01] *adaptability* is defined as a process which changes the behaviour of a software or system. Thereby, the desired functionality is preserved.

The field of application of the concept, which has been developed in context of this thesis, covers various domains, e.g. automotive industry, avionics or railway. However, different modelling notations or risk assessment procedures are used in the corresponding environments. Regardless of which modelling notation or risk assessment procedure is used the functionality of the developed approach has to be ensured. In this context, three parts are evaluated:

1. *System modelling:* Within the concept of this thesis system modelling is used to determine suitable alternative solutions for the MCDM. Furthermore, dependencies and commonalities are calculated in context of the SPL based MCDM. Finally, structural as well as KPI based impacts can be calculated based on SMs.

2. *Safety risk assessment:* This procedure is mandatory to estimate risks of alternative solutions depending on corresponding POVs preventatively. Thereby, risks are classified into different levels.

3. *Security risk assessment:* To protect a system against malicious attacks of third parties appropriate CMs are needed. Thereby, individual risks are assessed by means of the ADTA which is based on the ADT.

The approach which has been presented in this thesis uses the UML component diagram for system modelling. When adapting the type of system modelling, various modelling notations are available. These include, e.g. the Systems Modeling Language (SysML) from the Object Management Group (OMG) or the Architecture Analysis & Design Language (AADL). The functionality of the concept is being evaluated by means of a scenario which uses SysML block diagram as system modelling language instead of component diagrams. To derive alternative solutions from the component diagram it is essential that there are system components, ports and connections between them. This feature is supported by SysML block diagram. The block diagram uses system components and different types of relationships, e.g. to refine, derive or trace correlations between them. In this way, alternative solutions can be determined. Furthermore, components, ports and connections are needed to calculate dependencies and commonalities for semantic clustering of SPLs. The SysML block diagram provides a copy relationship which checks commonalities and thus dependencies from corresponding system components. Therefore, the SysML is suitable for this calculation. Since it is possible to

represent interrelations between different components a structural impact analysis based on component diagrams is enabled. To enable this feature for SysML block diagram the trace relationship has to be used. Components can be annotated with attributions and thus KPI based impact analysis can be realised. In this way, UML component diagrams can be replaced by SysML block diagrams.

The FMEA is used in the concept of this thesis to assess risks and to calculate trade-offs. It is now evaluated whether trade-offs can be calculated if the FMECA is used instead of the FMEA. In general, it is important that a measurable numerical value is available to assess the individual risks. For this purpose, the FMEA provides the RPN which can range between 1 and 1000. The MCDM approach of this thesis normalises the corresponding RPN value to calculate trade-offs. When applying the FMECA instead of the FMEA the risk is additionally classified into different criticality levels. Subsequently, if the FMECA classification is normalised the calculation of the trade-offs can be performed by means of RPN calculations as proposed in the concept of this thesis as well as by means of the criticality level.

The ADTA is performed by means of the ADT which mainly consists of the attack and the corresponding developed CMs. It is purpose of the ADTA to assess the individual CMs. In the following, it is evaluated whether the ADT can be replaced by EIDs to ensure correctness for the ADTA. The main difference is that the ADT is built top-down based on an attack which is refined to CMs, whereas the EID is built bottom-up based on CMs which prevent corresponding attacks. However, both of them provide possible CMs for individual attacks. Calculations for the ADTA require a set of CMs for an attack. This condition is fulfilled for both of them the ADT and EID.

Given the fact that all presented scenarios can be successfully applied the quality attribute *adaptability* (cf. Definition 6.1) is fulfilled for the approach of this thesis.

6.3.1.2. Scalability

The section evaluates whether the quality attribute scalability is fulfilled for selected scenarios. First, we define the term *scalability*:

Definition 6.2 (Scalability)**.** According to [TH19] *scalability* is defined as follows: Scalability is a property which describe the capability of a process or software system to "to grow and manage increased demand." [TH19] Scalability is often an indicator for stability and competitiveness of the software system since it increases productivity and handles a variety of belated demands.

The scope of the individual models presented within the concept of this thesis may grow over time independently of the respective domain. This includes the presented models SM, SSTM, ADT as well as the FM. In this context, modular sub-systems are often extended. Hereinafter, it is shown how far scalability may concern the individual model types:

1. *SM:* When extending a modular sub-system to a whole system, the underlying SM needs to be scaled accordingly to take necessary system components into account for the MCDM.
2. *SSTM:* If a sub-system is extended to a whole system it is mandatory to scale the SSTM as well. I.e., it is extended by additional goals and POVs to be able to calculate trade-offs in consideration of the whole system.
3. *ADT:* When scaling the SM and SSTM an ADT must also be scalable to develop and consider appropriate CMs for a security goal or POV within the SGH.
4. *FM:* In case of providing new functionality the corresponding FM needs to be extended and thus it has to be scalable. I.e., SPLs, which depend on the FM, have to be scalable.

Nowadays, final products in safety-critical environments are usually very complex, i.e. the underlying system structure is also very extensive. To keep the overview, however, sub-systems are first developed which are then combined to form a complete system. A scenario is now used to evaluate the extent to which modular sub-systems can be scaled to an overall system during system modelling. It is assumed that all these sub-systems are modelled correctly. As described in Section 3.3.2 the individual system components communicate via ports and interfaces. Conversely, i.e. the individual modular sub-systems are linked via ports and interfaces to form a consistent system. For instance, the individual SMs representing ACC, LA and LDP are summarised to an entire SM which considers all ADASs. In this way, several sub-systems can be scaled to a holistic system without loss of any functionality. The new alternative solutions are derived from the ADAS SM, i.e. sub-systems ACC, LA and LDP are taken into account which are essential to calculate corresponding trade-offs by means of the MCDM.

In the next step, the SSTM needs to scale as well. In this context, a root which represents the whole system has to be created. Each modular sub-system is a direct child node of this root node. For instance, a root goal node *ADAS is acceptably safe* is created which is refined by three further goals: *ACC is acceptably safe*, *LA is acceptably safe* and *LDP is acceptably safe*. Subsequently, it is necessary to create a comparison matrix of the root node as described in Section 2.4.1 and to weight the individual modular sub-systems against each other, i.e. local priorities for the goals *ACC/LA/LDP is acceptably safe* have to be determined. Since there are already pairwise comparisons for the sub-systems there are no further improvements needed. Finally, the risk assessment between alternative solutions and POVs has to be adapted to the scaled set of alternative solutions (cf. previous paragraph) if applicable to consider the entire set of solutions for the MCDM modes PCM (cf. Section 3.8.1) and RCM (cf. Section 3.8.2).

When scaling a SSTM it has to be considered that there are ADTs extensions with the purpose of developing appropriate CMs. These ADTs can also be scaled and

extended to provide CMs which are available for the entire system and not only for the individual modular sub-systems. For instance, CMs have to be developed which prevent security attacks of ADAS instead of ACC, LA or LDP standalone. This has no effects on the the calculation procedure of the ROI and ROA (cf. Section 3.7.2).

When realising SPLs it is necessary to create FMs to model functionality of the corresponding systems. Merging two or more modular sub-systems the procedure is quite similar to the SSTM. First, a root node is needed which summarises the entire functionality, e.g. the root feature *ADAS* is created. Subsequently, the individual sub-features of the sub-systems, e.g. *ACC*, *LA* and *LDP* including their relationships between them can be attached. In this way, the scaled FM can be used to apply MC in SPLs and to calculate trade-offs by means of the MCDM.

According to Definition 6.2 the presented scenario ensures full functionality, i.e. the quality attribute is fulfilled.

6.3.1.3. Reusability

This section evaluates the quality attribute reusability by means of a selected scenario. First, we define the term *reusability*:

Definition 6.3 (Reusability)**.** According to [SR90] *reusability* is defined as the reuse of software or hardware artefacts in different formats within the product development process.

The individual modular model components of the concepts of this thesis can be reused to reduce and save overhead. Only minor adjustments should be necessary to respond to current needs. Afterwards, it is analysed which kind of model types are involved by quality attribute reusability:

1. *SM:* If there are similar systems with minor changes, the SM can be reused with appropriate adjustments.
2. *SSTM:* There is a SCS with similar SST requirements with minor changes. In this case the SSTM can be reused with appropriate enhancements.
3. *ADT:* Analogue to SSTM there may be similar security vulnerabilities. Divergences has to be taken into account for development of CMs.
4. *FM:* If the requirements change slightly, an existing FM can be reused assuming that the corresponding FM is adapted.

The quality attribute *reusability* is evaluated by the following scenario: A system component of a system is replaced by another system component, e.g. a radar sensor is replaced when changing the manufacturer. This is checked by means of the developed change impact analysis. As a result the current concerned system

component needs to be removed including all ports, interfaces and connections between other system components. The replaced radar sensor system component has to be added and linked with corresponding system components via ports and interfaces, e.g. with the ACC system component. Since the replaced system component influences individual alternative solutions (cf. Section 3.3.3), the solutions has to be adapted accordingly. In this way, an existing system modelling can be reused by only adjusting minor and minimal changes.

Since there is a linking between system components and goals/POVs (cf. Figure 1.4), unused goals or POVs, which are accompanied by the replacement of the system component, can be removed. In this context, all the linked goals and POVs have to be reused for calculating trade-offs by means of the MCDM. Furthermore, it is essential to consider the replaced system component, e.g. the new radar system component. Therefore, corresponding goals and POVs may be needed which are individualised according to the replaced radar sensor, e.g. consideration of special weather conditions. Moreover, the risk assessment between POVs and alternative solutions may be adapted (cf. Section 3.7.1). The SSTM can be reused with a minimal adjustment.

Next, the ADT, which is interconnected with a security goal within the SSTM, can be reused. This exactly applies when the corresponding security goal is not affected by the replacement of the linked system component of the scenario. For instance, the security goal *Radar sensor acceptably secure against manipulation and hacking attacks* is always necessary, i.e. it applies to the replaced radar sensor as well. In this case, no further adjustments are necessary since a security goal is extended by exactly one ADT. I.e., the corresponding ADT with its developed CMs can be reused.

The FM can be reused as well. Since there is a linking between features and system components, features with an existing linking can be reused. For instance, the ACC feature is still provided by the radar sensor. Additionally, the FM is extended according to the replacement of the corresponding system component.

Thus it was shown that the developed concept of this thesis is reusable as defined in Definition 6.3.

6.3.1.4. Maintainability

The quality attribute maintainability is evaluated within this section. First, the term *maintainability* is defined:

Definition 6.4 (Maintainability)**.** According to [Dhi06] *maintainability* is defined as the probability that a system or parts of them can be repaired or changed within a given time frame that an operable state is restored.

The individual parts of the concepts of this thesis should be maintainable to guarantee functionality at any time. This should be done with minimal effort in a short time frame. In the following, the individual partial concepts are introduced with regard to maintainability:

1. *Multi-Concerns and MCDM:* This part is the basis for the concept of this thesis. I.e., all further parts of this thesis are based on it and therefore an easy maintenance is mandatory.

2. *Change Impact Analysis:* Impacts are clearly identifiable and traceable. Therefore, an effective maintenance of the procedure is of great interest.

3. *Multi-Concerns in SPLs:* It is not possible to check each SPL configuration with regard to SST requirements. For this purpose, an efficient maintenance has to be ensured to calculate trade-offs for SPLs in safety-critical context.

The quality attribute *maintainability* is evaluated by the following scenario: The construction of an underlying SM is enhanced due to failures regarding their structure. For instance, the communication between the radar sensor and an ECU has been set wrong, i.e. the SM needs to be maintained. As stated in the meta-model of Multi-Concerns and MCDM in Section 6.2.1 the SM is an essential part for performing the MCDM as the alternative solutions depend therefrom. Since the SSTM meta-model references the corresponding SM an update can be easily done by referencing the enhanced SM. In this way, the set of alternative solutions including FMEA risk assessments may be adapted according to structure changes of the SM if applicable. This exactly applies if the communication with the ECU is part of the alternative solutions. In this case, the FMEA assessments between alternative solutions and POVs have to be set again. Finally, the MCDM can be performed with the maintained SM and trade-offs can be calculated. Therefore, the first approach of this thesis supports maintainability.

When reviewing the meta-model of the change impact analysis (cf. Figure 6.4) the change impact analysis is triggered by an individual model element. The updated SM is mandatory to calculate an optimal trade-off, i.e. without updating the underlying SM impacts that concern the SSTM and thus the MCDM could not be taken into account or the results are falsified since there may be incorrect relationships. Consequently, maintaining the SM of the use case scenario considers the change impact analysis as well.

The approach, which concerns Multi-Concerns in SPLs, depends on system modelling. As described in Section 6.2.3 or illustrated in Figure 6.5 the FM, which is necessary for representing SPLs, is linked with the SM to get a mapping between feature $\leftrightarrow$ system component. Therefore, updating the SM of the introduced scenario influences maintainability and functionality of the approach is ensured. Otherwise, clustering of semantically equivalent features as proposed in Section 5.3.2 is calculated incorrectly. I.e., the subsequent MCDM calculates the trade-offs based

on insufficient SST goals.

In this way, it has been shown that maintainability as specified in Definition 6.4 is fulfilled for the selected use case scenario.

6.3.2. System under Development Evaluation

The scenario based evaluation, which has been covered in Section 6.3.1, aimed to evaluate the developed concept of this thesis. It has not been evaluated how far the SuD is influenced by the approach of this thesis. This part is the aim of this section.

6.3.2.1. Adaptability

There are a lot of reasons for adaptability, e.g. requirements change over time. Furthermore, supplier of individual system components may change. Moreover, it is possible to detect a more cost-effective alternative. For this reason, the SuD should be adaptable, i.e. impacts of change requests have to be determined correctly. Furthermore, the impacts have to be traceable as described in Objective 2 of Section 1.2. Afterwards, it is evaluated how far the change impact analysis is fulfilled for the selected scenario which is defined hereinafter: There is an existing airbag system of an automotive vehicle. In this context, the air pillows of the current system should be replaced by another air pillows of a competitive supplier. We evaluate the two different types of change impact analysis by means of the selected use case scenario with regard to the following properties:

1. *Structural impacts:* It is evaluated how far the calculated impact complies with the principle of traceability over the process chain.

2. *KPI based impacts:* It is also reviewed whether qualitative requirements with regard to SST are fulfilled.

When replacing an air pillow system component by another air pillow, which is identical in construction, the structural change impact analysis is triggered by a `mod` operation since the current system component of the underlying SM is modified. In this context, BC and WC analysis is performed, i.e. a minimum and maximum set of affected components is calculated for replacing the air pillow system component. As described in Section 4.3 modifying an underlying SM has subsequent impacts on the SM itself as well as the SSTM which is mainly responsible for calculating trade-offs by means of the MCDM. This means replacing, i.e. modifying an air pillow has effects on another system components, e.g. the trigger mechanism. Moreover, the replacement of the air pillow influences the calculation of trade-offs since the replacement of the air pillow may require an adaption of the solution set which is an essential part for calculating the corresponding trade-offs. For instance, the new air pillow may be only combined with individual gas mixture and has a maximum permissible inflation pressure. When applying

the corresponding impact rules of the concerned model types as described in Section 4.3.1 and Section 4.3.2 traceability is fulfilled due to the dependencies between the individual system components and alternative solutions. In this way, the entire chain of effects is identified if an air pillow is replaced.

Air pillows of an airbag have to fulfil SST requirements. In this context, usually the SILs have to be met. Furthermore, real time requirements have to be fulfilled since the air pillow is inflated with a gas mixture to protect vehicle occupants in case of an impact. Therefore, when replacing, i.e. modifying the air pillow, care has to be taken to ensure that elements with identical SST properties may be concerned as well by this process. In this way, the communication via bus system may be affected as well. The KPI based calculation of impacts as described in Section 4.5.2 identifies in this way goals and POVs which fulfil same SST requirements as the corresponding system components representing the air pillow to be replaced. Afterwards, the identified goals and POVs can be correctly adapted to get an unadulterated result of the MCDM. Neglecting the KPI based impacts may yield the realisation of the wrong trade-off and thus to delayed or faulty releasing the airbag in case of an emergency. It has been shown that adaptability of the SuD is possible by means of the presented approach of the change impact analysis.

It has been evaluated by means of a selected use case scenario that applying the structural and KPI based impacts fulfils the desired functionality completely.

6.3.2.2. Modularity

This section evaluates whether the quality attribute modularity is fulfilled for a SuD by means of the presented approach of this thesis. First, the term *modularity* is defined in detail:

Definition 6.5 (Modularity)**.** According to [BC06] *modularity* defines to which extent a software system or application is divided into smaller and more clear software construction modules. The process of merging single software modules is necessary to ensure functionality of the entire software system or application.

As described in Objective 3 of Section 1.2 safety-critical software systems have to fulfil variability in context of trade-off calculations. This aim is achieved by providing a variety of functionality and quality in SPLs (cf. Definition 5.1). In practice, companies develop independently of each other in different departments the individual functionality of the entire software system. For instance, an ACC is developed in another department than the LA. In this way, modularity of the SuD has to be supported to finally provide MCDM for safety-critical MC. In the following it is evaluated by means of a selected use case scenario whether modularity supports MC in SPLs: The individual ADASs ACC, LDP and LA are developed modularly and are merged to an ADAS functionality.

When developing a single ADAS, each ADAS has its own SM to model dependencies between individual hardware and software components. Since the result set of the MCDM depends on the underlying SM it is an essential part for calculating trade-offs. I.e., it is possible for every modular ADAS to calculate individual trade-offs. Therefore, the first property of *modularity* (cf. Definition 6.5) is fulfilled. Furthermore, modularity means to combine several standalone systems to guarantee functionality for the entire system according to the SPL configuration set. In this use case scenario the SPL configuration set corresponds to the ADASs as described in the previous paragraph. The SMs of the individual ADASs, i.e. ACC, LA and LDP have to be merged via port connections and interfaces. In this way, there is a result set of alternative solutions which considers the three ADASs. Consequently, the entire ADASs are taken into account when performing the MCDM which is customised for the SPL configuration. The calculation has to comply with the linkings and annotations as described in Section 5.2. Since the approach of this thesis can be applied to the modular systems of the scenario and is capable to combine and apply them to a SPL configuration set the quality attribute *modularity* as specified in Definition 6.5 is fulfilled.

6.3.2.3. Extensibility

It is evaluated by means of a selected scenario whether a SuD is extensible if the approach of this thesis is applied. However, we first define the term *extensibility* for understanding:

Definition 6.6 (Extensibility)**.** According to [Ber+95] *extensibility* is defined as the capability to extend a software (system) dynamically by new functionality. In this context, existing functionality of the software (system) has not to be impaired.

As stated in Objective 3 of Section 1.2 variability has to be ensured for SPL. To achieve this goal, it has to be possible to extend functionality at any time. In this context, functionality means that the calculation process of trade-offs by using MCDM is ensured and thus Objective 1 is fulfilled. In practice, functionality is extended by research and development in the corresponding domains. Hereinafter, the quality attribute extensibility is evaluated by the following use case scenario: An autonomous ADAS is released for general public. Thereby, the functionality has already been implemented but has not yet been activated due to regulations which have been decided by the government.

Since, the functionality has already been implemented but it has not activated the underlying SMs exist which are necessary for the basic operating principle. The modular SMs merely need to be merged to a entire software system. This part has already been evaluated in Section 6.3.2.2. Providing autonomous driving functionality requires extending the underlying FM to supply variability for the MC in SPLs. In this context, the extended features, i.e. autonomous driving package are accordingly linked with the SM and attributions and annotations

have to be completed as described in Section 5.2 to enable semantic clustering. The calculation of the extended functionality has to be possible independently of the already existing functionality to comply with the principle of SPLs and thus Objective 3. Obviously, the same applies for other functionality of the software system. The extended operations can be seen as modular elements since it is integrated within a SPL. Therefore, it has to work isolated as well as it may be part of the entire software system and functionality of the automotive vehicle. However, this part reflects the quality attribute *modularity* and has already been evaluated successfully in Section 6.3.2.2. I.e., the scenario can be extended by the approach of this thesis without any restrictions of functionality and the calculations of trade-offs (Objective 1) in consideration of individual SPLs (Objective 3) are fulfilled.

6.3.2.4. Maintainability

Maintainability as defined in Definition 6.4 is an essential part of software development process. Lack of maintainability leads to flaws in the final product which should be commercialised. Therefore, it is useful to develop single modular software modules to keep a better overview of the software development process of the final product. It has already been evaluated in Section 6.3.2.2 that a SuD fulfils modularity. Furthermore, a SuD has to be adaptable as needed to integrate belated changes flawless and with minimal effort. It has already been proven in Section 6.3.2.1 for a selected use case scenario that a SuD is adaptable when applying the approach of this thesis. Moreover, maintainability means that a software system is extensible to provide additional desired functionality. This part has been successfully evaluated in Section 6.3.2.3. Since the system to be applied fulfils modularity, adaptability and extensibility the quality attribute *maintainability* is fulfilled.

6.4. Case Study

It is the aim of this section to present two case studies by which the concept of this thesis is performed. The first case study covers a Turn Indicator (TI) example from an automotive vehicle. Thereby, the complete MCDM as proposed in Chapter 3 is applied. Moreover, the structural impact analysis is performed which is an essential part of Chapter 4. The second case study deals with an extensive ACC example from the automotive industry. In this case study the focus is on MC in SPLs as proposed in Chapter 5. Furthermore, the KPI based change impact analysis is performed for this selected case study.

6.4.1. Turn Indicator

Each vehicle needs to be equipped with a TI to participate in road traffic. However, flaws of TI may impair safety of road users. For instance, failure of emergency

indicators may lead to rear-end collision and thus endanger safety of passengers. It is therefore the aim of this section to perform the MCDM to determine the best hardware and software configuration of a TI. Moreover, it is evaluated which impacts the replacement of a necessary system component will have. Finally, it has to be mentioned that the TI case study is based on [Pel+11].

6.4.1.1. System Model

A TI has to fulfil several hardware and software requirements, which has to be realised by means of a SM. First, there is the flashing when opening or closing the car via the radio key. The crash or emergency flashing is responsible for concurrent flashing in case of an accident or congestion. Moreover, the turn flashing is activated when operating the lever to signal change of direction. In addition to the turn flashing there is the comfort flashing which activates turn flashing three times when softly applying the lever. Furthermore, theft flashing serves as a small protection against thieves. In this process, the flashing lamps light up at shorter intervals in combination with the horn. Usually commands are sent via bus infrastructure. In this case study, the focus is on the communication between the individual flashing modules. Therefore, we abstract from the bus infrastructure. Nevertheless, the communication between individual flashing modules has to be encrypted and thus a system component *Encryption* is needed which is regulated via the central component *Turn Indicator*. The underlying and associated SM is illustrated in Figure 6.7.

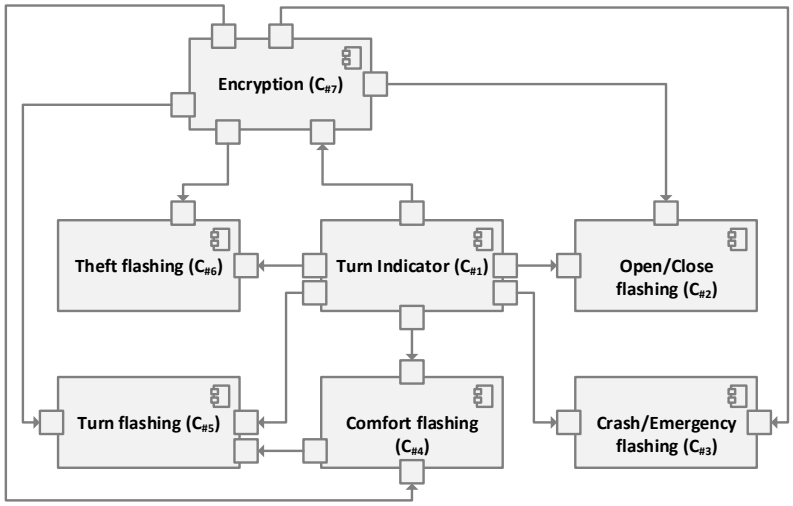


Figure 6.7.: SM of the TI case study

6.4.1.2. Trade-Offs and Counter Measures

As described in Section 3.4.1 to calculate an optimal trade-off we first need an appropriate SSTM. The SSTM which is illustrated in Figure 6.8 represents the root

goal that a *Turn Indicator (TI) is acceptably safe*. The root goal is refined by four further sub-goals which are responsible that the TI module, software and flashing is working correctly. Furthermore, it is important that the *TI communication is acceptably reliable*. Each of the four sub-goals is refined by at least two POVs which are essential to calculate trade-offs. The goal *TI module is working correctly* considers by means of POVs that failures are sufficiently mitigated if *TI lever returns to its starting position after steering movement* as well as if *Acoustic signal sounds when activating the Turn Indicator*. The goal *TI software is working correctly* ensures that results are correct without any failures. Furthermore, the results must be available in time and delays are sufficiently mitigated. Moreover, it is covered by a POV that the *Software is acceptably secure against manipulation and hacking attacks*. The goal which is responsible for correct flashing is refined by two further POVs in consideration of mitigating failures of affected lamps and flashing modules. The goal which is responsible for a reliable TI communication is refined by four POVs: It is considered that the communication is acceptably secured against data theft and manipulation. Moreover, importance is attached to the fact that messages arrive in time and are transferred correctly with acceptable reliability. Finally, there are two alternative solutions for which the trade-off is calculated by means of the MCDM. First, there is an alternative solution consisting of a basic TI functionality. Besides, there is a solution which additionally considers a comfort TI.

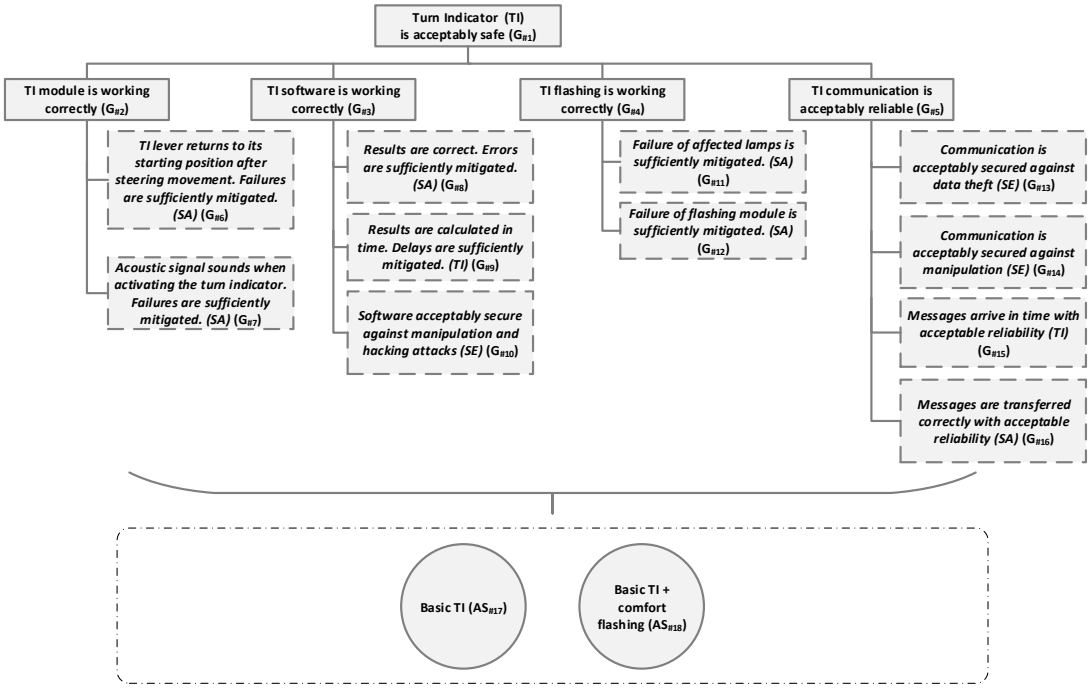


Figure 6.8.: SSTM of the TI case study

Subsequently, it is mandatory to perform the AHP algorithm for the goal hierarchy within the SSTM. I.e., a pairwise comparison as described in Section 2.4.1 has to

be performed based on $G_{\#1}$, $G_{\#2}$, $G_{\#3}$, $G_{\#4}$ and $G_{\#5}$. Since it is most important that flashing and the TI models are working correctly the resulting priorities for $G_{\#1}$ are $G_{\#4} \succ G_{\#2} \succ G_{\#5} \succ G_{\#3}$. Flashing has highest priority since flashing is the expected functionality. Software on the other hand has lowest priority since it must be designed redundantly. The probability is very low that all redundant software components do not calculate correct results. In case of $G_{\#2}$ it is more important that the TI returns to its starting position after steering movement than to sound an acoustic signal when activating the TI. The reason is that the functionality of the TI is given if no acoustic signal sounds. Therefore, the following applies for $G_{\#2}$: $G_{\#6} \succ G_{\#7}$. Furthermore, there is goal $G_{\#2}$ representing that TI software should work correctly. In this case it is most important that calculated results are correct since wrong results may lead to wrong direction indicator. The timing aspect is important but less important than calculating correct results. Usually, software components are designed redundantly, i.e. the probability is very low that all redundant software modules do not fulfil the timing constraints. In conclusion, the following applies for $G_{\#3}$: $G_{\#10} \succ G_{\#8} \succ G_{\#9}$. $G_{\#4}$ covers that TI flashing is working correctly. In the course of this, it is more important that failures of flashing modules are sufficiently mitigated than the failures of affected lamps. The reason is that lamps can not flash without corresponding flashing module. Obviously, the priorities are for $G_{\#4}$: $G_{\#12} \succ G_{\#11}$. When considering the local priorities of $G_{\#5}$ it is noticeable that it is most important that messages are transferred correctly. It is, however, less important that the communication is acceptably secured against data theft since in case of data theft functionality is still ensured in most cases. Therefore, the following local priorities apply for $G_{\#5}$: $G_{\#16} \succ G_{\#15} \succ G_{\#13} \succ G_{\#14}$. The exact distribution of the priorities including consistency ratios can be taken from Table 6.1 to Table 6.5.

$$G_{\#1} = \begin{array}{c} \\ G_{\#2} \\ G_{\#3} \\ G_{\#4} \\ G_{\#5} \end{array} \begin{array}{c} \begin{array}{cccc} G_{\#2} & G_{\#3} & G_{\#4} & G_{\#5} \end{array} \\ \begin{bmatrix} 1 & 3 & \frac{1}{4} & 3 \\ \frac{1}{3} & 1 & \frac{1}{4} & \frac{1}{2} \\ 4 & 4 & 1 & 3 \\ \frac{1}{3} & 2 & \frac{1}{3} & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#2}$	24,5 %
$G_{\#3}$	8,7 %
$G_{\#4}$	53,4 %
$G_{\#5}$	13,4 %
CR	**9,14 %**

Table 6.1.: AHP matrices and local priorities of the TI case study ($G_{\#1}$)

$$G_{\#2} = \begin{array}{c} \\ G_{\#6} \\ G_{\#7} \end{array} \begin{array}{c} \begin{array}{cc} G_{\#6} & G_{\#7} \end{array} \\ \begin{bmatrix} 1 & 4 \\ \frac{1}{4} & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#6}$	80,0 %
$G_{\#7}$	20,0 %
CR	**0,0 %**

Table 6.2.: AHP matrices and local priorities of the TI case study ($G_{\#2}$)

After calculating local priorities it is necessary to perform FMEA assessments as

proposed in Section 3.7.1. In this context, the RPN needs to be determined for the alternative solutions depending on the corresponding POVs. As a reminder: The RPN is the product of OSD as defined in Definition 2.7. The complete FMEA risk assessment of the TI case study is listed in Table 6.7. When comparing the RPN values between the two different alternative solutions it is conspicuous that $AS_{\#17}$ performs better than $AS_{\#18}$. Furthermore, it can be stated, that the assessments between the alternative solutions and the POVs, which belong to $G_{\#2}$, perform best whereas the POVs of $G_{\#5}$ have the worst assessments. The reason is that the communication and data transfer of the TI offers the most attack capabilities. $G_{\#3}$ and $G_{\#4}$ are in balance and ranked between $G_{\#2}$ and $G_{\#5}$ since the software and flashing cannot work without any communication whereas the TI module is almost independently of the communication.

$$G_{\#3} = \begin{array}{c} \\ G_{\#8} \\ G_{\#9} \\ G_{\#10} \end{array} \begin{array}{c} \begin{array}{ccc} G_{\#8} & G_{\#9} & G_{\#10} \end{array} \\ \begin{bmatrix} 1 & 6 & 3 \\ \frac{1}{6} & 1 & \frac{1}{3} \\ \frac{1}{3} & 3 & 1 \end{bmatrix} \end{array}$$

Goal	Local Priority
$G_{\#8}$	65,5 %
$G_{\#9}$	9,5 %
$G_{\#10}$	25,0 %
CR	**1,74 %**

Table 6.3.: AHP matrices and local priorities of the TI case study ($G_{\#3}$)

$$G_{\#4} = \begin{array}{c} \\ G_{\#11} \\ G_{\#12} \end{array} \begin{array}{c} \begin{array}{cc} G_{\#11} & G_{\#12} \end{array} \\ \begin{bmatrix} 1 & \frac{1}{3} \\ 3 & 1 \end{bmatrix} \end{array}$$

Goal	Local Priority
$G_{\#11}$	25,0 %
$G_{\#12}$	75,0 %
CR	**0,0 %**

Table 6.4.: AHP matrices and local priorities of the TI case study ($G_{\#4}$)

$$G_{\#5} = \begin{array}{c} \\ G_{\#13} \\ G_{\#14} \\ G_{\#15} \\ G_{\#16} \end{array} \begin{array}{c} \begin{array}{cccc} G_{\#13} & G_{\#14} & G_{\#15} & G_{\#16} \end{array} \\ \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{5} & \frac{1}{7} \\ 3 & 1 & \frac{1}{3} & \frac{1}{5} \\ 5 & 3 & 1 & \frac{1}{3} \\ 7 & 5 & 3 & 1 \end{bmatrix} \end{array}$$

Goal	Local Priority
$G_{\#13}$	5,5 %
$G_{\#14}$	11,8 %
$G_{\#15}$	26,2 %
$G_{\#16}$	56,5 %
CR	**4,41 %**

Table 6.5.: AHP matrices and local priorities of the TI case study ($G_{\#5}$)

All the prerequisites are now fulfilled to calculate trade-offs by means of the MCDM as described in Chapter 3. I.e., MCDM modes as proposed in Section 3.8 can be applied. The results can be taken from Table 6.6. It is noted that the PCM uses an even distribution of OSD whereas the PCM (modified) ranks severity better than occurrence and detection. However, no matter which algorithm is applied, $AS_{\#17}$ is the better trade-off than $AS_{\#18}$. The results can be justified as follows: As defined in Table 6.1 $G_{\#4}$ has the highest priority. Furthermore, the

FMEA assessments between the associated POVs $G_{\#11}$ or $G_{\#12}$ and the alternative solutions are better for $AS_{\#17}$. Consequently, the $AS_{\#17}$ is better fulfilled when applying the PCM. $AS_{\#17}$ fulfils the modified PCM to a greater extent since the assessments for severity are identical for both alternative solutions, but $AS_{\#17}$ has better assessments for occurrence and detection than $AS_{\#18}$. $AS_{\#17}$ complies with RCM better than $AS_{\#18}$ since each RPN is better for $AS_{\#17}$ than for $AS_{\#18}$.

MCDM mode	$AS_{\#17}$	$AS_{\#18}$
RCM	60,7 %	39,3 %
PCM	59,8 %	40,2 %
PCM (modified)	55,9 %	44,1 %

	O	**S**	**D**	**Local Priority**
O	1	$\frac{1}{3}$	1	20,0 %
S	3	1	3	60,0 %
D	1	$\frac{1}{3}$	1	20,0 %

Table 6.6.: Results of the TI case study

When analysing the RPNs of Table 6.7 it is noticeable that the RPN of $G_{\#10}$ has the worst assessments for both, $AS_{\#17}$ and $AS_{\#18}$. Therefore, we evaluate by means of the ADTA whether there is an appropriate CMs to mitigate the risk. I.e., we assess the ROI and ROA of corresponding CMs. The underlying ADT which should mitigate $G_{\#10}$ is illustrated in Figure 6.9. There are two types of attacks to manipulate the software of a car: Gaining violent and non-violent access to the car. In this context there are three attacks possible for the violent access in the following order: *Break down the door*, *Exploit vulnerability* and *Go out unobserved*. For the non-violent access the attack order may be: *Spy the key*, *Open the car with the key*, *Exploit vulnerability* and *Go out unobserved*. For each attack there is at least one CM which should avoid the risk of an attack. These CMs can be taken from Figure 6.9. It is now evaluated which of the nine CMs has the greatest benefit to enable a higher degree of security for the SSTM goal *Software acceptably secure against manipulation and hacking attacks*.

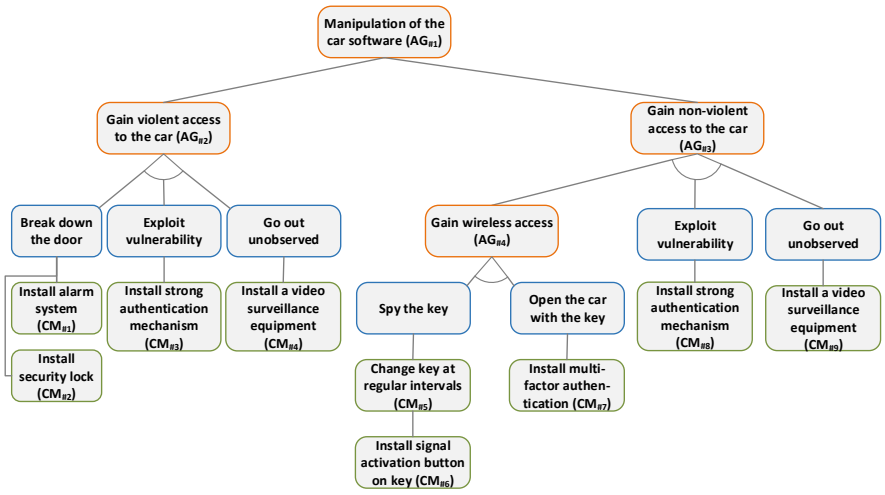


Figure 6.9.: ADT of the TI case study

OSD	$G_{\#6}$	$G_{\#7}$	$G_{\#8}$	$G_{\#9}$	$G_{\#10}$	$G_{\#11}$	$G_{\#12}$	$G_{\#13}$	$G_{\#14}$	$G_{\#15}$	$G_{\#16}$
$AS_{\#17}$	3/3/2	2/2/3	3/5/2	4/6/2	4/4/6	4/4/4	3/6/3	2/6/6	2/7/6	2/7/3	2/5/7
$AS_{\#18}$	2/3/3	2/2/5	5/5/3	4/6/4	4/4/6	5/4/4	3/6/4	2/6/6	2/7/6	3/7/4	2/5/7
RPN	$G_{\#6}$	$G_{\#7}$	$G_{\#8}$	$G_{\#9}$	$G_{\#10}$	$G_{\#11}$	$G_{\#12}$	$G_{\#13}$	$G_{\#14}$	$G_{\#15}$	$G_{\#16}$
$AS_{\#17}$	18	12	30	48	96	64	54	72	84	42	70
$AS_{\#18}$	18	20	75	96	96	80	72	72	84	84	70
Local Priority	$G_{\#6}$	$G_{\#7}$	$G_{\#8}$	$G_{\#9}$	$G_{\#10}$	$G_{\#11}$	$G_{\#12}$	$G_{\#13}$	$G_{\#14}$	$G_{\#15}$	$G_{\#16}$
$AS_{\#17}$	50,0 %	54,2 %	57,5 %	55,6 %	50,0 %	51,9 %	52,4 %	50,0 %	50,0 %	55,7 %	50,0 %
$AS_{\#18}$	50,0 %	45,8 %	42,5 %	44,4 %	50,0 %	48,1 %	47,6 %	50,0 %	50,0 %	44,3 %	50,0 %

Table 6.7.: FMEA risk assessment of the TI case study

As defined in Definition 3.6 calculation of ROI requires some other variables. These include AV, EF, ARO and CSI. The AV is assigned to 70.000 € for the TI case study. The EF is set to 85 % for $AG_{\#2}$, 90 % for $AG_{\#4}$ and 95 % for $AG_{\#3}$. I.e., the loss on the asset value is greatest when realising non-violent access to the car since there is more expensive equipment installed. The ARO is set 0,1 for $AG_{\#2}$; 0,2 for $AG_{\#4}$ and 0,3 for $AG_{\#3}$. Subsequently it is mandatory to determine RM for each CM which can be taken from Table 6.8. Thereby, $CM_{\#1}$ and $CM_{\#2}$ has the best risk mitigation for the violent access whereas $CM_{\#6}$ mitigates the risk to best degree in case of non-violent access. If the first step of a violent access is prevented the entire attack can be prevented with a higher probability. For the non-violent access it is most secure to activate signal activation button on the key on demand. This ensures that the key signals are not spied out at any undesired moment. Finally, we need to determine the costs for security investment for each CM. These vary from 1.500 € in case of $CM_{\#5}$ and $CM_{\#6}$ up to 3.000 € in case of $CM_{\#1}$, $CM_{\#4}$ and $CM_{\#9}$. If the formula as defined in Definition 3.6 is applied for each CM, $CM_{\#6}$ has the best value. The reason is that the risk mitigation as well as the costs for security investment are best. Furthermore, EF and ARO are on good average.

	$CM_{\#1}$	$CM_{\#2}$	$CM_{\#3}$	$CM_{\#4}$	$CM_{\#5}$	$CM_{\#6}$	$CM_{\#7}$	$CM_{\#8}$	$CM_{\#9}$
RM	0,8	0,8	0,6	0,35	0,4	0,8	0,7	0,6	0,35
CSI (€)	3000	2800	2700	3000	1500	1500	2000	2700	3000
Cost (€)	1000	1000	2000	500	3000	3000	3000	2000	500
Loss (€)	3500	5000	3500	2000	3000	5000	5000	5000	2000
ROI	3,76	4,10	2,97	1,08	2,36	**5,72**	3,41	3,43	1,33
ROA	11,11	8,33	9,10	20,00	8,33	**6,25**	**6,25**	7,14	20,00

Table 6.8.: ADTA of the TI case study

As stated in Definition 3.7 the ROA requires three variables to apply the formula. These include GI, $Cost$ and $Loss$. In case of a successful attack for an attacker the GI is assigned to 50.000 €. The values of $Cost$ and $Loss$ can be taken from Table 6.8. The higher the $Cost$ and $Loss$ the better the resulting ROA. In our TI case study $CM_{\#6}$ and $CM_{\#7}$ are the most expensive CMs for attackers to overcome them. When comparing the results of ROI and ROA $CM_{\#6}$ fulfils ROI and ROA best, i.e. it is recommended to realise this CM.

6.4.1.3. Structural Change Impact Analysis

The last part of the TI case study covers the structural change impact analysis with the aim of traceability. It is evaluated by means of a change request which kind of impacts it has when modifying the encryption ($C_{\#7}$, cf. Figure 6.7). As proposed in Section 4.3 it is distinguished BC and WC change impacts, i.e. there is a minimal and maximal set of effects. The structural change impact analysis is therefore triggered by $C_{\#7}$ which represent the encryption system component.

Since this system component is linked with the individual flashing modules, in the first iteration $C_{\#2}$, $C_{\#3}$, $C_{\#4}$, $C_{\#5}$ and $C_{\#6}$ need to be modified in WC. In BC there are no further effects. When continuing the WC structural change impact analysis the *Turn flashing* system component is affected in all further iterations by the component *Comfort flashing* with `mod`. As proposed in Section 3.3.2, alternative solutions are derived from the SM, i.e. the impacts of the individual system components affect the alternative solutions of the corresponding SSTM (cf. Figure 6.8). The necessary modification of *Turn flashing* has effects on both alternative solutions $AS_{\#17}$ and $AS_{\#18}$ since turn flashing is the basic functionality of TI. $AS_{\#18}$ is affected by *Comfort flashing* system component. All the applied impact rules are listed in Table 6.9.

Source/Target element	BC	WC
SM → SM	$C_{\#7}$.mod → $C_{\#2}$.noChange	$C_{\#7}$.mod → $C_{\#2}$.mod
	$C_{\#7}$.mod → $C_{\#3}$.noChange	$C_{\#7}$.mod → $C_{\#3}$.mod
	$C_{\#7}$.mod → $C_{\#4}$.noChange	$C_{\#7}$.mod → $C_{\#4}$.mod
	$C_{\#7}$.mod → $C_{\#5}$.noChange	$C_{\#7}$.mod → $C_{\#5}$.mod
	$C_{\#7}$.mod → $C_{\#6}$.noChange	$C_{\#7}$.mod → $C_{\#6}$.mod
SM → SSTM		$C_{\#4}$.mod → $AS_{\#18}$.mod
		$C_{\#5}$.mod → $AS_{\#17}$.mod
		$C_{\#5}$.mod → $AS_{\#18}$.mod

Table 6.9.: Impact rules of structural impacts of the TI case study

6.4.2. Adaptive Cruise Control

In the age of advancing autonomous driving the ACC is an essential part therefrom. As with the TI, flaws of the ACC may impair safety of road users. For instance, delayed braking of the ACC may lead to a life-threatening rear-end collision. In practice, there are several variants of an ACC supporting SPLs. Therefore, it is the aim of this section to apply the MCDM on a safety-critical SPL as proposed in Chapter 5. Furthermore, a KPI based change impact analysis is performed as described in Section 4.6. For the most part, the ACC case study is based on [LFB18] and [LB19].

6.4.2.1. System Model and Feature Model

An ACC must fulfil several hardware and software requirements which must be specified within the underlying SM. There is a system component *Sensor technology* which covers installed and used sensor types of an ACC, e.g. radar or video camera. The *actuator technology* is responsible for accelerating or decelerating of the corresponding automotive vehicle. Furthermore, the components *Engine control* and *Brake control* regulate internal process with regard to throttling and braking. Both of them need the input of sensor and actuator technology. Moreover, the *Brake control* and *Engine Control* interact with each other, i.e. they are interconnected. To

adapt the maximum permissible speed of the ACC a *Cruise switch* is needed. In general, all commands of the ACC are sent via a bus architecture. This case study is focused on the communication between the individual presented modules of the ACC. Therefore, we abstract from the bus architecture. To communicate between the individual modules a central system component *Adaptive Cruise Control* is required. Furthermore, it is necessary to transmit the data using encryption algorithms. Therefore, a system component *Encryption* is used which is used by all other system components. The facts of the SM described in this paragraph are also illustrated in Figure 6.10.

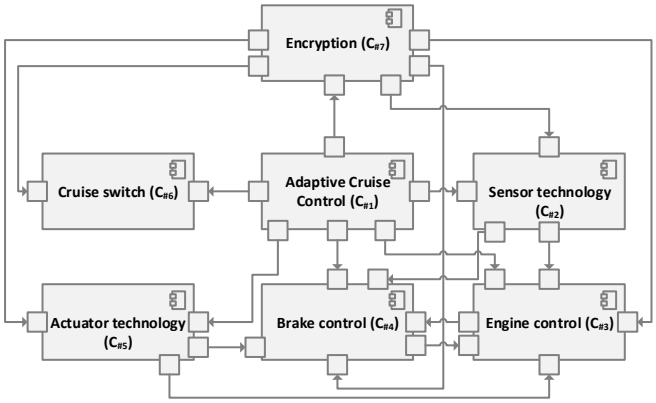


Figure 6.10.: SM of the ACC case study

All the functional features, which represent the collectivity of all available SPLs, are specified in Figure 6.11. There is an abstract feature *Adaptive Cruise Control* which supports four abstract and concrete features. First, it is distinguished between two cruising speeds, *160 km/h* and *210 km/h*. In this process, only one of them can be selected for the SPL configuration. Furthermore, there is an optional feature *Congestion assistance* which accomplishes accelerating and braking up to 30 km/h. Moreover, a mandatory dynamic distance control can be configured either by three or five available levels. Finally, a dynamic speed limit is possible, i.e. the maximum permissible speed is either registered via Global Positioning System (GPS) or via a road sign recognition. It should be noted that the dynamic speed limit via road sign recognition is only configurable if a cruising speed of 210 km/h is selected.

6.4.2.2. SST Model and FMEA

To calculate trade-offs for individual SPLs we first need a SSTM which is presented in this section. There is a root goal that the *ACC is acceptably safe*. This root goal is refined by four essential sub-goals. First, a sub-goal covers that *ACC sensors are working correctly*. Furthermore, it is considered that the *ACC software is working correctly*. Moreover, the ACC actuators have to work correctly which are responsible for accelerating and braking. A reliable ACC communication is

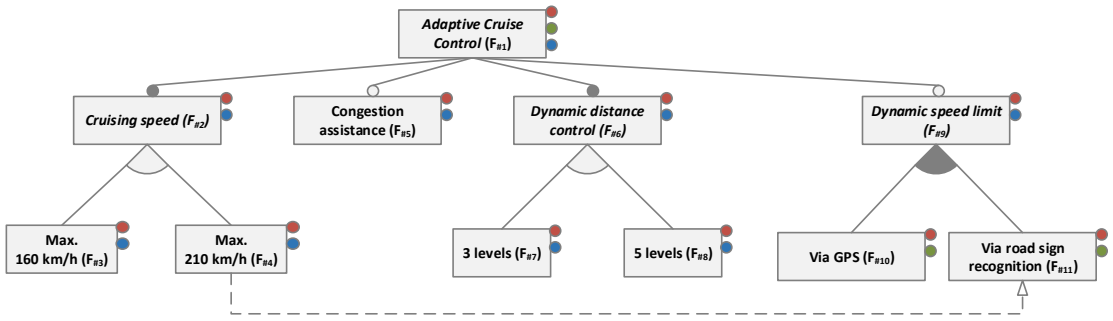


Figure 6.11.: FM of the ACC case study

an essential part of the correct functionality of the entire ACC system. Each of the sub-goals has to be fulfilled by at least two POVs. If the correct functionality of the ACC sensors is considered it has to be taken into account that *missing an obstacle can be ruled out with sufficient certainty*. Furthermore, care has to be taken to ensure that distance to an obstacle, e.g. a vehicle driving ahead is acceptably accurate. The goal *ACC software is working correctly* is refined by two POVs. The first one ensures that the software is acceptably secure against manipulation and hacking attacks. The second POV covers calculating results in time. If there are any delays it is sufficiently mitigated. When considering goals that an ACC actuator is working correctly it has to be taken into account that brake failure and engine failure is sufficiently mitigated. The ACC communication goal requires that the corresponding communication is secured against data theft and manipulation. Furthermore, it is necessary that the individual messages arrive in time and thus are reliable. Within the SSTM alternative solutions are defined which are essential for calculating trade-offs. In general, they are derived from the underlying SM. Since there is a linking between FM and SM, the corresponding features are considered in the alternative solutions. Hereinafter, the alternative solutions are presented with the following features:

1. Cruising speed: 160 km/h, dynamic distance control: 3 levels, dynamic speed limit: GPS
2. Cruising speed: 210 km/h, congestion assistance, dynamic distance control: 5 levels, dynamic speed limit: road sign recognition
3. Cruising speed: 160 km/h, congestion assistance, dynamic distance control: 5 levels, dynamic speed limit: GPS

The complete SSTM in graphical representation is illustrated in Figure 6.12.

As usual for every MCDM, it is mandatory to rank the individual underlying goals or POVs by each other. In the course of the root goal the local priorities of the sub-goals are ranked as: $G_{\#4} \succ G_{\#2} \succ G_{\#5} \succ G_{\#3}$. Since the actuator performs braking and accelerating it is most important whereas software has lowest priority

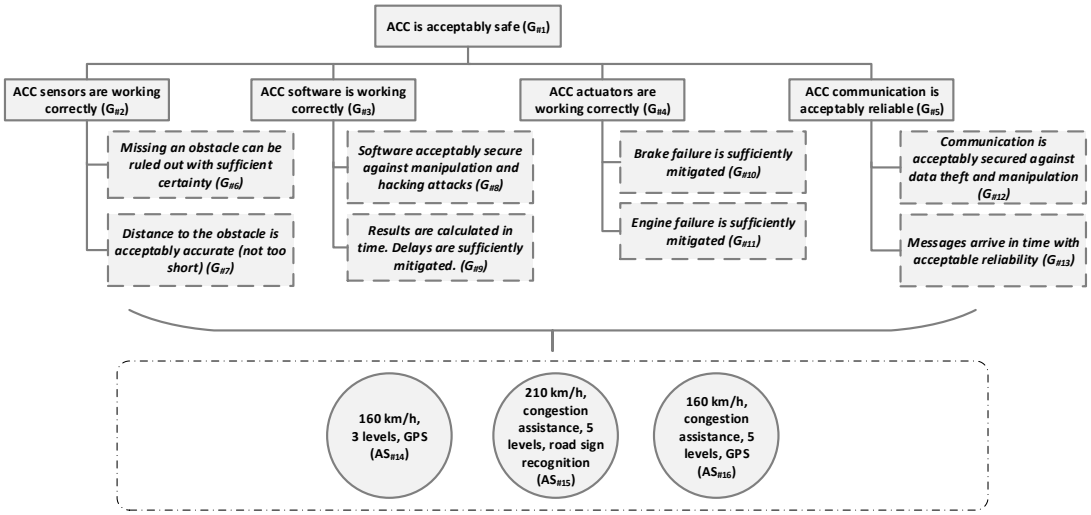


Figure 6.12.: SSTM of the ACC case study

since braking and accelerating should work without software support in emergency case. The exact values can be taken from Table 6.10.

$$G_{\#1} = \begin{array}{c} \\ G_{\#2} \\ G_{\#3} \\ G_{\#4} \\ G_{\#5} \end{array} \begin{array}{c} \begin{array}{cccc} G_{\#2} & G_{\#3} & G_{\#4} & G_{\#5} \end{array} \\ \begin{bmatrix} 1 & 3 & \frac{1}{4} & 3 \\ \frac{1}{3} & 1 & \frac{1}{4} & \frac{1}{2} \\ 4 & 4 & 1 & 3 \\ \frac{1}{3} & 2 & \frac{1}{3} & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#2}$	24,5 %
$G_{\#3}$	8,7 %
$G_{\#4}$	53,4 %
$G_{\#5}$	13,4 %
CR	**9,14 %**

Table 6.10.: AHP matrices and local priorities of the ACC case study ($G_{\#1}$)

When comparing the individual POVs considering sensors by each other the following applies: $G_{\#7} \succ G_{\#6}$. It is, in general, most important to keep distance to the vehicle driving ahead. Furthermore, in case of ACC software it is most important that the results are calculated in time since the probability of hacking attacks is also possible but lower than timing aspects. In detail, it applies: $G_{\#9} \succ G_{\#8}$. In case of actuators it is more important that brake failures are sufficiently mitigated than to mitigate engine failures. The reason is that functioning brakes prevents accidents in higher grade than the engine. Obviously, the following applies: $G_{\#11} \succ G_{\#10}$. When comparing the POVs of goal $G_{\#5}$ it has to be noted that it is more important that messages arrive in time with acceptable reliability than the communication is secured against data theft and manipulation. The justification for this is similar to that for $G_{\#3}$. Consequently, the local priorities are: $G_{\#13} \succ G_{\#12}$. The individual local priorities can be taken from Table 6.11 to Table 6.14.

Subsequently, it is necessary to determine FMEA assessments for the alternative

solutions $AS_{\#14}$, $AS_{\#15}$ and $AS_{\#16}$ depending on the individual POVs. The detailed RPNs can be taken from Table 6.15. $AS_{\#14}$ has the best RPNs for the POVs concerning ACC actuators since it has the least configurations. $AS_{\#15}$ has the maximum configuration, consequently the software for the alternative solution has to work properly. $AS_{\#16}$ has best RPN values for ACC actuators since it only supports a speed limit of 160 km/h with three installed sensor types.

$$G_{\#2} = \begin{array}{c} \\ G_{\#6} \\ G_{\#7} \end{array} \begin{array}{c} \begin{array}{cc} G_{\#6} & G_{\#7} \end{array} \\ \begin{bmatrix} 1 & \frac{1}{4} \\ 4 & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#6}$	20,0 %
$G_{\#7}$	80,0 %
CR	**0,0 %**

Table 6.11.: AHP matrices and local priorities of the ACC case study ($G_{\#2}$)

$$G_{\#3} = \begin{array}{c} \\ G_{\#8} \\ G_{\#9} \end{array} \begin{array}{c} \begin{array}{cc} G_{\#8} & G_{\#9} \end{array} \\ \begin{bmatrix} 1 & \frac{1}{5} \\ 5 & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#8}$	16,7 %
$G_{\#9}$	83,3 %
CR	**0,0 %**

Table 6.12.: AHP matrices and local priorities of the ACC case study ($G_{\#3}$)

$$G_{\#4} = \begin{array}{c} \\ G_{\#10} \\ G_{\#11} \end{array} \begin{array}{c} \begin{array}{cc} G_{\#10} & G_{\#11} \end{array} \\ \begin{bmatrix} 1 & 6 \\ \frac{1}{6} & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#10}$	85,7 %
$G_{\#11}$	14,3 %
CR	**0,0 %**

Table 6.13.: AHP matrices and local priorities of the ACC case study ($G_{\#4}$)

$$G_{\#5} = \begin{array}{c} \\ G_{\#12} \\ G_{\#13} \end{array} \begin{array}{c} \begin{array}{cc} G_{\#12} & G_{\#13} \end{array} \\ \begin{bmatrix} 1 & \frac{1}{4} \\ 4 & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#12}$	20,0 %
$G_{\#13}$	80,0 %
CR	**0,0 %**

Table 6.14.: AHP matrices and local priorities of the ACC case study ($G_{\#5}$)

RPN	$G_{\#6}$	$G_{\#7}$	$G_{\#8}$	$G_{\#9}$	$G_{\#10}$	$G_{\#11}$	$G_{\#12}$	$G_{\#13}$
$AS_{\#14}$	54	60	36	60	24	24	48	72
$AS_{\#15}$	48	60	18	24	96	96	32	48
$AS_{\#16}$	48	60	36	60	32	32	48	72

Table 6.15.: FMEA assessments of the ACC case study

When applying the MCDM without consideration of SPLs the global priorities (regardless of the selected MCDM mode) are: $AS_{\#14} \succ AS_{\#16} \succ AS_{\#15}$. Since $G_{\#4}$ has the highest local priority for $G_{\#1}$ (cf. Table 6.10) and $AS_{\#14}$ has the best FMEA assessment for $G_{\#10}$ and $G_{\#11}$, $AS_{\#14}$ is the best trade-off and $AS_{\#15}$ is the worst alternative solution. The detailed results can be taken from Table 6.16.

	RCM	**PCM**
$AS_{\#14}$	43,0 %	36,7 %
$AS_{\#15}$	23,0 %	29,2 %
$AS_{\#16}$	34,1 %	34,1 %

Table 6.16.: Results of the conventional MCDM of the ACC case study

6.4.2.3. Software Product Lines and Trade-Offs

The last section covered a conventional MCDM. However, no SPLs as proposed in Chapter 5 have been taken into account. It is the aim of this section to continue the ACC case study considering SPLs. As a reminder, some prerequisites have to be fulfilled (cf. Section 5.3) to enable such a MCDM:

1. There is a SM, FM and SSTM.
2. Next, there is a linking between the individual features of the FM and the system components of the SM.
3. Furthermore, there is a linking between features of the FM and the individual POVs of the SSTM.
4. Commonly used system components are determined for the selection of alternative solutions.
5. The features are annotated with SST flags to determine the individual concerns.
6. The KPI attribution for all the POVs is specified.
7. The FMEA assessments of the individual POVs are clustered.
8. Finally, a bottom-up path formation within the SSTM can be performed.

The SM, FM and SSTM have already been introduced in the two sections before. Therefore, we continue with the linkings between FM, SM and SSTM which are listed in Table 6.17.

For the linkings between FM and SM only the concrete features are relevant since abstract features are not selectable. If a cruising speed of 160 km/h or 210 km/h is chosen the system component *Cruise switch* is responsible within the SM. Furthermore, the congestion assistance is mapped to *Sensor technology* since radar

FM	SM	SSTM
$F_{\#3}$	$C_{\#6}$	$G_{\#6}$, $G_{\#7}$, $G_{\#8}$, $G_{\#10}$, $G_{\#11}$, $G_{\#12}$
$F_{\#4}$	$C_{\#6}$	$G_{\#6}$, $G_{\#7}$, $G_{\#8}$, $G_{\#10}$, $G_{\#11}$, $G_{\#12}$
$F_{\#5}$	$C_{\#2}$	$G_{\#8}$, $G_{\#10}$, $G_{\#11}$
$F_{\#7}$	$C_{\#2}$	$G_{\#6}$, $G_{\#7}$, $G_{\#8}$
$F_{\#8}$	$C_{\#2}$	$G_{\#6}$, $G_{\#7}$, $G_{\#8}$
$F_{\#10}$	$C_{\#5}$	$G_{\#9}$, $G_{\#13}$
$F_{\#11}$	$C_{\#5}$	$G_{\#9}$, $G_{\#13}$

Table 6.17.: Linkings between FM, SM and SSTM of the ACC case study

sensors control the functionality. The three levels dynamic distance control uses primary the same system component since it is driven by radar sensors as well. If there is a dynamic speed limit via GPS or road sign recognition the functionality is very demanding, i.e. the features are linked with the system component *Actuator technology*.

When using a cruising speed of 160 km/h or 210 km/h within a SPL, we need to link all the POVs which are responsible for the sensor technology. It has to be ensured that the software is secured against manipulation and hacking attacks. Furthermore, it has to be considered that brake failure is sufficiently mitigated. Moreover, a link with all the communication POVs is necessary to enable a reliable communication. In case of the congestion assistance feature we also need a software which is secured against third parties. In addition, all the actuators have to fulfil functional scope properly. When linking the concrete features of the dynamic distance control it is essential that the ACC sensors are working correctly. Furthermore, it is advisable to link it with the POV which is responsible for secure software and against manipulation and hacking attacks. Finally, the dynamic speed limits require a linking with two POVs: The first one covers timing aspects in context of ACC software. The second one concerns ACC communication, in detail it ensures that messages arrive in time with acceptable reliability. In this context the timing aspect is very important since speed limit is only valid for a specific road section.

For this case study, a SPL is selected consisting of a maximum cruising speed of 160 km/h, a dynamic distance control of three levels and a dynamic speed limit via GPS. In this way, alternative solution $AS_{\#16}$ is supported for the subsequent MCDM. However, as stated in step 1 of Section 5.3.2 we need to check regarding commonly used system components. In this process Table 6.18 in cooperation with Table 6.17 may be helpful.

It is stated that $C_{\#2}$ is used by the features $F_{\#5}$, $F_{\#7}$ and $F_{\#8}$ which are part of $AS_{\#14}$ and $AS_{\#15}$. I.e., all three alternative solutions are taken into account for calculating the optimal trade-off.

Solution	Supported features
$AS_{\#14}$	$F_{\#3}$, $F_{\#7}$, $F_{\#10}$
$AS_{\#15}$	$F_{\#4}$, $F_{\#5}$, $F_{\#8}$, $F_{\#11}$
$AS_{\#16}$	$F_{\#3}$, $F_{\#5}$, $F_{\#8}$, $F_{\#10}$

Table 6.18.: Supported features of individual solutions of the ACC case study

When calculating trade-offs by means of SPLs it is necessary to minimise the underlying SSTM to reduce complexity for the analysis process (cf. Section 5.3.2). For this purpose, we need to determine SST flags for each feature defined in the FM. Only features with identical SST annotations can be summarised to the same cluster. The SST annotations of the ACC case study can be taken from Figure 6.11 (for legend: cf. Figure 2.9). In general, the ACC supports SST concern. For the cruising speed safety and security are important since manipulating of the correct cruising speed may endanger human life. The same applies to the congestion assistance. If the ACC does not respond with the correct braking or accelerating amount road users are also in danger. The dynamic distance control also reflects safety and security aspects since a manipulation of the distance may lead to serious rear-end collisions. In case of dynamic speed limit safety and timing are significant. If a speed limit road sign is recognised too late with unacceptable delay it may cause serious accents. Since there is a linking between FM and SSTM (cf. Table 6.17) the SST annotations of the features can be transferred logically to the POVs of the SSTM.

Since SST annotations are not enough to form clusters we need to define attributions for each POV within the SSTM. These attributions are listed in Table 6.19. In this case study for each POV two KPIs are specified. In this context, SST attributions are taken into account. There is the *SIL* which is used to assess safety POVs. Furthermore, *encryption* is used to ensure a higher degree of security. Finally, *time* indicates whether time limits must be observed. KPIs can be eliminated for the calculation of trade-offs if the SST flags and the corresponding annotated KPIs are identical. This applies to $G_{\#8}$ and $G_{\#12}$ as well as $G_{\#9}$ and $G_{\#13}$, i.e. one of each cluster can be eliminated. The reason is as follows: *KPI 1* and *KPI 2* are identical for $G_{\#8}$ and $G_{\#12}$ as well as for $G_{\#9}$ and $G_{\#13}$.

Furthermore, a safety and security annotation is assigned to $G_{\#8}$ and $G_{\#12}$ whereas a safety and timing flag is assigned to $G_{\#9}$ and $G_{\#13}$. Table 6.20 shows the individual clusters with the corresponding related POVs. The optimisation is finished if each cluster consists of one POV, i.e. the eliminated POVs have not be taken into account for the calculation of trade-offs.

In the next step we have to decide which of the similar POVs are eliminated for the calculation of the trade-offs. For this purpose, we compare $G_{\#8}$ and $G_{\#12}$ as well as $G_{\#9}$ and $G_{\#13}$ regarding classifications and average RPNs. It is mandatory that the classifications of the RPNs are identical. Otherwise they would deviate too much

POV	KPI 1	KPI 2
$G_{\#6}$	*(sil, min, 3)*	*(fmea, max, 50)*
$G_{\#7}$	*(sil, min, 3)*	*(fmea, max, 60)*
$G_{\#8}$	*(encryption, min, 128)*	*(fmea, max, 60)*
$G_{\#9}$	*(time, max, 150)*	*(fmea, max, 60)*
$G_{\#10}$	*(sil, min, 2)*	*(fmea, max, 96)*
$G_{\#11}$	*(sil, min, 2)*	*(fmea, max, 100)*
$G_{\#12}$	*(encryption, min, 128)*	*(fmea, max, 60)*
$G_{\#13}$	*(time, max, 150)*	*(fmea, max, 60)*

Table 6.19.: KPIs of the ACC case study

Cluster	Related POVs
$Cl_{\#1}$	$G_{\#6}$
$Cl_{\#2}$	$G_{\#7}$
$Cl_{\#3}$	$G_{\#8}$, $G_{\#12}$
$Cl_{\#4}$	$G_{\#9}$, $G_{\#13}$
$Cl_{\#5}$	$G_{\#10}$
$Cl_{\#6}$	$G_{\#11}$

Table 6.20.: Clusters of the ACC case study

and, if applicable, cause new risks . As listed in Table 6.21 the classifications are identical. Consequently, $G_{\#8}$ has the better RPN ranking in comparison with $G_{\#12}$. Moreover, $G_{\#9}$ compared with $G_{\#13}$ has the better RPN values. For this reason $G_{\#12}$ and $G_{\#13}$ is eliminated for the calculation of the trade-offs.

Classification	$G_{\#8}$	$G_{\#9}$	$G_{\#12}$	$G_{\#13}$
$AS_{\#14}$	Acceptable	Medium	Acceptable	Medium
$AS_{\#15}$	Acceptable	Acceptable	Acceptable	Acceptable
$AS_{\#16}$	Acceptable	Medium	Acceptable	Medium
∅ **RPN**	**30**	**48**	43	64

Table 6.21.: Average RPNs and classifications of the ACC case study

In this way, the POVs $G_{\#6}$, $G_{\#7}$, $G_{\#8}$, $G_{\#9}$, $G_{\#10}$ and $G_{\#11}$ are used for the calculation of trade-offs. All the paths up to the root goal $G_{\#1}$ are enabled. I.e., the local priorities of $G_{\#1}$ are updated since $G_{\#5}$ has been eliminated due to $G_{\#12}$ and $G_{\#13}$. Afterwards, the updated local priorities of $G_{\#1'}$ are listed:

$$G_{\#1'} = \begin{array}{c} \\ G_{\#2} \\ G_{\#3} \\ G_{\#4} \end{array} \begin{array}{c} \begin{array}{ccc} G_{\#2} & G_{\#3} & G_{\#4} \end{array} \\ \begin{bmatrix} 1 & 3 & \frac{1}{4} \\ \frac{1}{3} & 1 & \frac{1}{5} \\ 4 & 5 & 1 \end{bmatrix} \end{array}$$

Goal	**Local Priority**
$G_{\#2}$	22,6 %
$G_{\#3}$	10,1 %
$G_{\#4}$	67,4 %
CR	**8,14 %**

Table 6.22.: AHP matrices and local priorities of the ACC case study ($G_{\#1'}$)

Finally, the MCDM can be performed. The results can be taken from Table 6.23. Thereby, $AS_{\#14}$ has the best trade-off whereas $AS_{\#15}$ has the worst trade-off. When comparing these results with those of Table 6.16 it can be determined that the results differ minimally from each other. In this way it can be determined that the complexity reduction returns the same ranking of the individual alternative solutions. However, considering SPLs for the calculation of trade-offs saves time, costs and resources. According to Definition 5.2 there is a complexity reduction of 25 %.

	RCM	**PCM**
$AS_{\#14}$	43,4 %	36,9 %
$AS_{\#15}$	21,5 %	28,4 %
$AS_{\#16}$	35,1 %	34,7 %

Table 6.23.: Results of SPL based MCDM of the ACC case study

6.4.2.4. KPI Based Change Impact Analysis

The TI case study covered a structural change impact analysis. In this section KPI based change impact analysis is performed on the already known ACC case study. First, the SM, which has been introduced in Section 6.4.2.1, is annotated with at least one KPI. The mappings between the individual system components and KPIs can be taken from Table 6.24.

System component	**KPI 1**	**KPI 2**
$C_{\#1}$	-	-
$C_{\#2}$	*(sil, min, 3)*	*(encryption, max, 128)*
$C_{\#3}$	*(sil, min, 1)*	*(encryption, min, 64)*
$C_{\#4}$	*(sil, min, 3)*	*(encryption, min, 64)*
$C_{\#5}$	*(sil, min, 2)*	*(encryption, max, 128)*
$C_{\#6}$	*(sil, min, 3)*	*(encryption, max, 128)*
$C_{\#7}$	*(encryption, min, 64)*	*(time, max, 150)*

Table 6.24.: KPIs of the ACC System Model

In general, there are KPIs which specify that a certain SIL has to be reached. Thereby, the *Sensor technology*, *Brake control* and *Cruise switch* require the strictest

demands. Furthermore, security KPIs have to be considered by means of key length of encryption. In this context the key length vary 64 bit up to 128 bit. For instance, the sensor technology can require a 128 bit encryption. There are also strict timing requirements which must be taken into account. In general, the encryption may not exceed timing frames of 150 ms.

In this case study it is evaluated which elements of the individual models impair the calculation of trade-offs if a change request is performed. Hereinafter, the change request is defined (cf. Section 4.5.2):

1. The radar sensor is replaced by another radar sensor, i.e. the system component *Sensor technology* is replaced by another one.
2. The new radar sensor have some new attribution triples:
   a) (sil, min, 4) and
   b) (encryption, min, 256).
3. *allAttributes = true*
4. $\Delta_{FMEA} = (0, -1, -1)$

When comparing with the current radar sensor the new sensor technology has better safety and security demands. The change impact analysis takes effects if both KPI triples are not fulfilled for the corresponding elements. If the analysis affects some POVs the severity and detection decreases by 1 when performing the FMEA at a later stage.

When performing the KPI based change impact analysis the following elements are affected by the sensor replacement:

1. SM: $C_{\#6}$, $C_{\#8}$, $C_{\#10}$.
2. FM: $F_{\#3}$, $F_{\#4}$, $F_{\#5}$, $F_{\#7}$, $F_{\#8}$.
3. SSTM: $G_{\#6}$, $G_{\#7}$, $G_{\#8}$, $G_{\#10}$, $G_{\#11}$, $G_{\#12}$.

Since the current installed sensor technology requires a SIL of 3 and a maximum key length of 128 bit with regard to encryption only $C_{\#6}$ share these demands. As described in Section 6.4.2.3, the SM is linked with the FM and SSTM (cf. Table 6.17). Therefore, the corresponding affected features and POVs can be determined. In summary, the cruising speed of 160 km/h and 210 km/h, the congestion assistance as well as the dynamic distance control with three and five levels are affected by the radar sensor replacement.

In the following, let us assume the radar sensor and all affected elements has been replaced correctly and we want to perform the MCDM based on SPLs again as described in Section 6.4.2.3. To do this, the corresponding RPNs (cf. Table 6.15) are updated automatically according the rule Δ_{FMEA} = (0, -1, -1). The updated RPN values can be taken from Table 6.25.

RPN	$G_{\#6}$	$G_{\#7}$	$G_{\#8}$	$G_{\#9}$	$G_{\#10}$	$G_{\#11}$	$G_{\#12}$	$G_{\#13}$
$AS_{\#14}$	30	32	16	60	12	12	30	72
$AS_{\#15}$	30	32	8	24	60	60	18	48
$AS_{\#16}$	30	32	16	60	18	18	30	72

Table 6.25.: Improved FMEA assessments of the ACC case study

Subsequently, it has to be checked again whether there is a change in the elimination of POVs. The average value of $G_{\#8}$ and $G_{\#12}$ have changed but the classification has not changed. The updated values are listed in Table 6.26.

Classification	$G_{\#8}$	$G_{\#9}$	$G_{\#12}$	$G_{\#13}$
$AS_{\#14}$	Acceptable	Medium	Acceptable	Medium
$AS_{\#15}$	Acceptable	Acceptable	Acceptable	Acceptable
$AS_{\#16}$	Acceptable	Medium	Acceptable	Medium
∅ **RPN**	**13**	**48**	26	64

Table 6.26.: Updated average RPNs and classifications of the ACC case study

Finally, the MCDM algorithm can calculate the global priorities for the individual alternative solutions once again. The actual results does not change, i.e. the following applies: $AS_{\#14} \succ AS_{\#16} \succ AS_{\#15}$. The percentage distribution has changed a little bit. This has to do with the fact that local priorities of the individual alternative solutions vary in the updated process of the FMEA assessment. The updated results are listed in Table 6.27.

	RCM	**PCM**
$AS_{\#14}$	45,9 %	44,8 %
$AS_{\#15}$	20,2 %	20,6 %
$AS_{\#16}$	33,9 %	34,6 %

Table 6.27.: Results of updated SPL based MCDM of the ACC case study

7 Conclusion and Outlook

In the previous chapters several approaches have been presented which deal with the compliance of the safety concern taken SST into account. These approaches have been evaluated qualitatively and two extensive case studies have been performed to show the benefits and advantages in MC engineering techniques. In this chapter, the thesis and their results are summarised and a final statement is made whether the objectives, which have been defined in Section 1.2, were accomplished. Furthermore, a short outlook and improvements are given how future works could be realised to continue the research work of this thesis.

7.1. Summary

In this thesis, three approaches on MC techniques have been introduced. These include Multi-Concerns and Multi-Criteria Decision Making, the change impact analysis as well as Multi-Concerns in Software Product Line. These concepts have been consecutively developed where each approach aims to improve the antecedent algorithmic to finally provide an optimal overall approach. In this context, each of the proposed approaches aimed to solve the objectives which have been presented in Section 1.2 of this thesis.

In general, all the approaches or objectives aim to guarantee a maximum degree of safety in context of safety-critical systems. However, each of them presented different methodologies to achieve the goal. In the first approach, the focus was on calculating optimal trade-offs by means of MCDM taken SST concerns into account. The change impact analysis, which was part of the second main chapter, aimed to determine structural and KPI based effects triggered by a change request. Finally, the third concept focused on calculating trade-offs by enabling modelling of SPLs.

7.1.1. Multi-Concerns and Multi-Criteria Decision Making

In Chapter 3, an approach has been presented, which allows calculating trade-offs taken SST into account, to solve mutual conflicts of the individual concerns. In this context, a systems engineering process has been introduced which is a necessary prerequisite to perform the MCDM. As requested in Objective 1, defining functional and qualitative requirements including failure modes and goals formed the

basis to trigger the process. Based on these requirements, SMs are needed to model the dependencies of the SuD. From this, in turn, the sets of alternative solutions, as also required in Objective 1, can be derived for which the final trade-off is calculated.

Subsequently, it has been described that knowledge of failures modes and goals have to be transferred into a hierarchical structure to receive more accurate results of the MCDM. In this context, the SSTM has been presented which additionally includes modelling of the alternative solutions. As an extension the ADT has been described which allows modelling of CMs for individual security attacks. Without modelling of SSTM and ADT appropriate trade-offs as well as CMs cannot be finished. Since Objective 1 requires modelling of MC this part is also covered by the proposed approach.

Furthermore, it has been explained how to apply the AHP on the hierarchical structure of the SSTM. In this way, it has been described that it is mandatory for the calculation of trade-offs to carry out pairwise comparisons for all the goals modelled within the SSTM. It enables determining the relative importance or local priorities of the individual goals by means of comparison matrices. In this context, an algorithm for improving consistency has been presented since it is mandatory to achieve an appropriate level of transitivity.

Moreover, two techniques for risk assessment or risk mitigation as demanded in Objective 1 have been presented for the calculation of trade-offs. The first one is the FMEA, which assess the risk of individual alternative solutions depending on so-called POV by means of the RPN. In this way, probabilities of three multipliers have been considered: Occurrence, severity and detection. Furthermore, the ADTA has been proposed which is based on the ADT. By means of probability values and cost values the efficiency of CMs is evaluated which can be applied to mitigate risk.

Finally, two algorithms have been proposed for the calculation of the final results. In this way, a percentage distribution between the individual previously specified alternative solutions is calculated. The RCM and PCM algorithm, which have been presented at the end of Chapter 3, are mainly based on the local priorities of the AHP and the FMEA risk assessment. Furthermore, an optimisation of the PCM is possible when adapting the local priorities of the multipliers of the FMEA.

In summary, Objective 1 has been covered the approach which has been proposed in Chapter 3. On the one hand MC are modelled within the SSTM. On the other hand, risk assessment and mitigation has been taken into account to guarantee a maximum degree of safety or security. The final trade-off calculation is done by means of realising the AHP.

7.1.2. Change Impact Analysis

Chapter 4 presented an approach which enables calculating effects based on triggering change requests. As required in Objective 2 horizontal traceability has been realised by developing individual change impact rules which differentiate between BC and WC, i.e. a minimum and maximum set of affected elements. In this context, all presented model types are involved including SM, SSTM, ADT and FM (topic of Chapter 5). In this way, the entire effect chain can be retraced as also demanded in Objective 2.

To retrace the whole effect chain, as required in Objective 2, it is necessary to release the change impact analysis. For this purpose, it has been presented how to define a change request. First, a stakeholder is needed to trigger a change request. Moreover, a requirement change has to be enjoined, e.g. due to new regulations. The change request has to be triggered by an affected model element, e.g. a goal and a triggering operation, e.g. modify.

To combine change requests and developed impact rules algorithms have been developed to fulfil Objective 2. First, a methodology has been developed to enable horizontal traceability over the entire effect chain based on previously specified change requests. In this case, the results of the impacts are differentiated between BC and WC. To avoid cycles, that may occur, a prioritisation of the applied change impact rules has been introduced. Furthermore, Objective 2 had the demand to guarantee still a maximum degree of safety in the course of the change impact analysis. Therefore, a second methodology, a KPI based technique has been presented. Based on attribution annotations of the release element items with identical qualitative requirements are identified.

In summary, Objective 2 has been realised in the course of Chapter 4 since traceability is enabled when combining change requests and impact rules. Furthermore, qualitative aspects have been considered when using the attributed impact methodology.

7.1.3. Multi-Concerns in Software Product Lines

In Chapter 5, an approach has been presented which enables calculating trade-offs based on SPL specifications taken SST requirements into account. As demanded in Objective 3 it is required to model functional variability to model SPLs. Furthermore, it was required to combine SPL techniques with Objective 1, i.e. a linking between FM and SM as well as SSTM is mandatory to select the alternative solutions and POVs or goals according to the SPL configuration set. In this way, the demand of Objective 3 for variability modelling and support has been fulfilled.

Moreover, it has been required in Objective 3 to cluster issues with similar SST requirements to reduce complexity and to enable reusability. To achieve this demand,

the commonly used system components within the SM have to be determined. Moreover, it was the need to annotate features within the FM regarding SST concerns. Furthermore, for the actual semantic clustering a KPI based attribution of the individual POVs is needed. Finally, the FMEA classification of the alternative solutions within the SSTM is necessary to complete the semantic clustering.

In Chapter 5, the change impact rules which have been introduced in Chapter 4 have been extended by rules which enable a change impact analysis for MCDM taken SPLs into account. In this way, horizontal traceability is possible as required in Objective 2. Furthermore, the entire effect chain is retraceable by using structural change impact methodology. In this context, four impact rule types have been added which concern FM, SM and SSTM. In this way, the change impact analysis supports variability management.

In summary, Objective 3 has been fulfilled when realising the concept of Chapter 5 since variability modelling via FMs is possible. Furthermore, reducing overhead for the calculation of trade-offs is realised by the semantic clustering algorithm. Finally, Objective 2 has been fulfilled since the presented approach of Chapter 5 has been extended by corresponding change impact rules.

7.2. Future Work

The approaches which have been presented in this thesis can be applied in safety-critical domains, e.g. the automotive industry. Especially, when using SPLs the models increase and are not easy to handle. For instance, the pairwise comparison matrices for more than five sub-goals or POVs can no longer be set manually since the quality with regard to consistency or transitivity underneath otherwise. For future works it might be useful to modularise systems into smaller sub-systems to reduce overhead. In this way, the individual models are getting smaller but no necessary details are neglected as a result. In this context, the first step should be to calculate trade-offs for individual sub-systems. Based on those results of the modular sub-systems the trade-offs will be determined for the overall system by merging the partial results. In that case, priorities of the individual sub-systems have to be set to follow up on the AHP and the MCDM in general.

Furthermore, it might be useful for future work to insert empirical values for FMEA, AHP, ADT assessments. For that purpose, a big database needs to exist for which the individual RPN values, pairwise comparisons, ROI and ROA multiplier have been set. The values could be taken from the database if a combination of FMEA, AHP or ADT assessments has been used for calculation of trade-offs beforehand. Moreover, the entries in individual databases may be labelled how far they contribute to get an optimal trade-off and thus solve conflicts. In this way, the results of the MCDM can be improved and thus safety of the SuD is getting better and better.

The testability of the concept of this thesis may be considered in future works. For this purpose, the work of Christian Saad [Saa15] which covers model-based data flow analysis should be integrated. It might be helpful to define invariants for the FMEA assessments as well as for the AHP pairwise comparisons. This ensures that no invalid values are used for the calculation of trade-offs. Moreover, the data flow analysis allows to trace local priorities of the corresponding goals or POVs and to compare them with the results of the final MCDM.

Part IV.

Annexe

Bibliography

[AE15] *The evolution of car safety: a history*. `https://www.autoexpress.co.uk/car-news/90221/the-evolution-of-car-safety-a-history`. accessed on June 25th, 2019. 2015.

[Ans+11] Saoussen Anssi et al. "Enabling Scheduling Analysis for AUTOSAR Systems". In: *2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing* (2011).

[Arn96] Robert S. Arnold. *Software Change Impact Analysis*. Los Alamitos, USA, 1996.

[BC06] Carliss Y. Baldwin and Kim B. Clark. "Modularity in the Design of Complex Engineering Systems". In: *Complex Engineered Systems: Science Meets Technology*. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2006, pp. 175–205.

[BCP02] G. Bernat, A. Colin, and S. M. Petters. "WCET Analysis of Probabilistic Hard Real-Time Systems". In: *23rd IEEE Real-Time Systems Symposium, 2002. RTSS 2002.(RTSS)*. 2002, p. 279.

[Ben+08] Nelly Bencomo et al. "Reflective Component-based Technologies to Support Dynamic Variability". In: *VaMos*. 2008.

[Ber+09] Bernd Bertsche et al. *Zuverlässigkeit mechatronischer Systeme, Grundlagen und Bewertungen in frühen Entwicklungsphasen*. Springer-Verlag Berlin-Heidelberg, 2009.

[Ber+95] B. N. Bershad et al. "Extensibility Safety and Performance in the SPIN Operating System". In: *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*. New York, USA, 1995, pp. 267–283.

[BFP06] S. Bistarelli, F. Fioravanti, and P. Peretti. "Defense trees for economic evaluation of security investments". In: *First International Conference on Availability, Reliability and Security (ARES'06)*. 2006.

[BLO03] L. C. Briand, Y. Labiche, and L. O'Sullivan. "Impact analysis and change management of UML models". In: *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.* 2003, pp. 256–265.

[BMI17] Bundesministerium des Inneren. "Fehlermöglichkeits- und einflussanalyse (FMEA)". In: *Organisationshandbuch* (2017).

[Böc+04] Günter Böckle et al. *Software-Produktlinien - Methoden, Einführung und Praxis*. Heidelberg, Germany: dpunkt.verlag, 2004, pp. 13–24.

[Boh02] S. A. Bohner. "Software change impacts-an evolving perspective". In: *International Conference on Software Maintenance, 2002. Proceedings*. 2002, pp. 263–272.

[Bow03] J.B. Bowles. "An assessment of RPN prioritization in a failure modes effects and criticality analysis". In: *Annual Reliability and Maintainability Symposium* (2003), pp. 380–386.

[Bra+12] Rosana T. Vaccare Braga et al. "Adapting a Software Product Line Engineering Process for Certifying Safety Critical Embedded Systems". In: *Computer Safety, Reliability, and Security*. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2012, pp. 352–363.

[But07] Bettina Buth. *Efficient Safety Analysis through Combination of Methods*. Hamburg, Germany, 2007.

[Che+11] C. Cheng et al. "An AHP Method for Road Traffic Safety". In: *Fourth International Joint Conference on Computational Sciences and Optimization (CSO)*. Dalian, China, 2011.

[CHE04] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. "Staged Configuration Using Feature Models". In: *Software Product Lines*. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2004, pp. 266–283.

[Coh94] Paul Cohn. *Elements of Linear Algebra*. CRS Press, 1994.

[CP12] *CESAR Project Homepage*. `http://www.cesarproject.eu/`. accessed on June 25th, 2015. 2012.

[Dhi06] B. S. Dhillon. *Maintainability, Maintenance, and Reliability for Engineers*. Tylor & Francis Group, 2006.

[DLM13] H. Mayela Delgado, Francisca Losavio, and Alfredo Matteo. "Goal oriented techniques and methods: Goal refinement and levels of abstraction". In: *2013 XXXIX Latin American Computing Conference (CLEI)* (2013).

[DOD49] United States Department of Defense. *MIL-P-1629: Procedures for Performing a Failure Mode, Effects and Criticality Analysis*. USA, 1949.

[EF19] Eclipse Foundation. *Sirius Architecture Overview*. `https://www.eclipse.org/sirius/doc/developer/Architecture_Overview.html`. accessed June 10th, 2019. 2019.

[ES09] Mathias Ekstedt and Teodor Sommestad. "Enterprise Architecture Models for Cyber Security Analysis". In: *2009 IEEE/PES Power Systems Conference and Exposition*. 2009, pp. 1–6.

[Eva86] Leonard Evans. "The Effectiveness of Safety Belts in Preventing Fatalities". In: *Accid. Anal. & Prev.* 18.3 (1986).

[EWL10] Franz Eisenführ, Martin Weber, and Thomas Langer. *Rationales Entscheiden*. Springer-Verlag Berlin-Heidelberg, 2010.

[Fen16] Andrea Fendt. "Modeling and Analyzing Safety, Security and Real-Time Requirements of Embedded Systems". MA thesis. Augsburg, Germany: University of Augsburg, 2016.

[Fer86] Olaf von Fersen. *Ein Jahrhundert Automobiltechnik: Personenwagen*. Düsseldorf, Germany: VDI-Verlag GmbH, 1986.

[FGS05] Benjamin Fabian, Oliver Günther, and Sarah Spiekermann. "Security Analysis of the Object Name Service". In: *Proc. First IEEE Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing* (2005).

[FIM10] Simone Fischer-Hübner, Luigi Lo Iacono, and Sebastian Möller. "Usable Security und Privacy". In: *Datenschutz und Datensicherheit - DuD* (2010).

[Fis+05] Kathi Fisler et al. "Verification and Change-impact Analysis of Access-control Policies". In: *Proceedings of the 27th International Conference on Software Engineering*. New York, USA, 2005, pp. 196–205.

[FK18] Reza Fattahi and Mohammed Khalilzadeh. "Risk evaluation using a novel hybrid method based on FMEA, extended MULTIMOORA, and AHP under fuzzy environment". In: *Safety Science* 102 (2018), pp. 290–300.

[Ger+10] David Gerónimo et al. "Survey of Pedestrian Detection for Advanced Driver Assistance Systems". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.7 (2010).

[GJ15] P. Göhner and N. Jazdi. *Zuverlässigkeit und Sicherheit von Automatisierungssystemen*. University of Stuttgart, Germany, 2015.

[GV17] V. B. Gisin and E. S. Volkova. "On Transitivity of Fuzzy Preferences Generated by Utility Functions with a Random Threshold". In: *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*. 2017, pp. 734–736.

[Har+14] Mark Harmann et al. "Search Based Software Engineering for Software Product Line Engineering - A Survey and Directions for Future Work". In: *Proceedings of the 18th International Software Product Line Conferewnce - Volume 1*. 2014, pp. 5–18.

[Har87] P. Harker. "Derivatives of the perron root of a positive reciprocal matrix: With application to the analytic hierarchy process". In: *Applied Mathematics and Computation* 22 (1987).

[HK18] Hitesh and A. Charan Kumari. "Feature Selection Optimization in SPL using Genetic Algorithm". In: *Procedia Computer Science* (2018), pp. 1477–1486.

[HMW11] Damien Hutchinson, Heath Maddern, and Jason Wells. "An Agile IT Security Model for Project Risk Assessment". In: *Proceedings of the 9th Australian Information Security Management Conference*. 2011.

[HSS05] A. Hanemann, D. Schmitz, and M. Sailer. "A framework for failure impact analysis and recovery with respect to service level agreements". In: *2005 IEEE International Conference on Services Computing (SCC'05) Vol-1*. 2005.

[IEC05] International Electrotechnical Commission (IEC). *Information technology – Security techniques – Information security management systems – Requirements*. norm. 2005.

[IEC16] International Electrotechnical Commission (IEC). *IEC 61882: 2016 Hazard and operability studies (HAZOP studies) - Application guide*. norm. 2016.

[IEC97] International Electrotechnical Commission (IEC). *Functional safety of electrical/electronic/programmable electronic safety-related systems*. standard. 1997.

[Jan10] Klaus Janschek. *Systementwurf mechatronischer Systeme: Methoden - Modelle - Konzepte*. Springer, 2010.

[Ji+10] X. Ji et al. "AHP implemented Security Assessment and Security Weight Verification". In: *Second International Conference on Social Computing (SocialCom)*. Leeds, UK, 2010.

[Jia+09] G. Jianbin et al. "The safety detection research of wind power units based on AHP method". In: *International Conference on Sustainable Power Generation and Supply*. Nanjing, China, 2009.

[Kaj99] M. Kajko-Mattsson. "Maintenance at ABB (II): Change Execution Processes (The State of Practice)". In: *Proceedings of the International Conference on Software Maintenance*. 1999, pp. 307–315.

[KH10] Haklin Klimm and Ho-sang Ham. "Integrated Fault Tolerant System for Automotive Bus Networks". In: *Computer Engineering and Applications (ICCEA)* (2010).

[Kli16] Bernd Kling. *Tesla Model S: Sicherheitsforscher hacken Elektroauto aus der Ferne*. `http://www.zdnet.de/88279165/tesla-model-s-sicherheitsforscher-hacken-elektroauto-aus-der-ferne/`. 2016.

[KW04] Tim Kelly and Rob Weaver. "The Goal Structuring Notation - A Safety Argumentation Notation". In: *Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases* (2004).

[KW18] Barbara Kordy and Wojciech Wideł. "On Quantitative Analysis of Attack–Defense Trees with Repeated Labels". In: *Principles of Security and Trust*. Springer International Publishing, 2018, pp. 325–346.

[LB19] Philipp Lohmüller and Bernhard Bauer. "Software Product Line Engineering for Safety-critical Systems". In: *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD*. Prague, Czech Republic, 2019, pp. 211–218.

[LDL07] Jing Liu, Josh Dehlinger, and Robyn Lutz. "Safety analysis of software product lines using state-based modeling". In: *Journal of Systems and Software* (2007).

[LFB18] Philipp Lohmüller, Andrea Fendt, and Bernhard Bauer. "Multi-Concerns Engineering for Safety-Critical Systems". In: *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD*. Funchal, Portugal, 2018, pp. 504–510.

[Li+18] Mole Li et al. "A Product Line Systems Engineering Process for Variability Identification and Reduction". In: *ArXiv e-prints* (2018).

[LPP11] Peter Löw, Roland Pabst, and Erwin Petry. *Funktionale Sicherheit in der Praxis - Anwendung von DIN EN 61508 und ISO/DIS 26262 bei der Entwicklung von Serienprodukten*. dpunkt.verlag Heidelberg, 2011.

[LRB19] Philipp Lohmüller, Julia Rauscher, and Bernhard Bauer. "Failure and Change Impact Analysis for Safety-Critical Systems". In: *Business Modeling and Software Design*. Lisbon, Portugal, 2019, pp. 47–63.

[LSB15] Melanie Langermeier, Christian Saad, and Bernhard Bauer. "Adaptive Approach for Impact Analysis in Enterprise Architectures". In: *Business Modeling and Software Design*. 2015, pp. 22–42.

[LT69] M. Stuart Lynn and William Timlake. "Bounds for Perron Eigenvectors and Subdominant Eigenvalues of Positive Matrices". In: *Linear Algebra and Its Applications* (1969), pp. 143–152.

[LYL16] Hu-Chen Liu, Jian-Xin You Yi-Zeng Chen, and Hui Li. "Risk evaluation in failure mode and effects analysis using fuzzy digraph and matrix approach". In: *Journal of Intelligent Manufacturing* 27 (2016).

[Mäk00] Minna Mäkäräinen. "Software change management processes in the development of embedded software". PhD thesis. University of Oulu, Finland, 2000.

[MB17] Mercedes-Benz Germany. *Benz-Patent-Motorwagen*. `https://www.mercedes-benz.com/de/mercedes-benz/classic/museum/benz-patent-motorwagen/`. accessed on November 22nd, 2019. 2017.

[MB87] Anna L. Martensen and Ricky W. Butler. "The Fault-Tree Compiler". In: *NASA Technical Memorandum 89098*. Hampton, Virginia, USA, 1987.

[McG06] Gary McGraw. *Software Security: Building Security In*. Boston, USA: Addison-Wesley Professional, 2006, pp. 4–10.

[Met+07] A. Metzger et al. "Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis". In: *15th IEEE International Requirements Engineering Conference (RE 2007)*. 2007, pp. 243–253.

[MLY05] Suvda Myagmar, Adam J. Lee, and William Yurcik. "Threat Modeling as a Basis for Security Requirements". In: *Symposium on Requirements Engineering for Information Security (SREIS)*. 2005.

[MO06] Sjouke Mauw and Martijn Oostdijk. "Foundations of Attack Trees". In: *Information Security and Cryptology - ICISC 2005*. Springer Berlin Heidelberg, 2006, pp. 186–198.

[MP17] *MERgE project homepage*. `http://www.merge-project.eu/project.html`. accessed on June 25th, 2019. 2017.

[Nea03] Richard E. Neapolitan. *Learning Bayesian Networks*. Artificial Intelligence, 2003.

[Nil96] Lena Nilsson. *Safety Effects of Adaptive Cruise Controls in Critical Traffic Situations*. Swedish National Road and Transport Research Institute, 1996.

[Ola07] Stefan Olander. "Stakeholder impact analysis in construction project management". In: *Construction Management and Economics* (2007), pp. 277–287.

[Ord+09] Lisa D. Ordóñez et al. "Goals Gone Wild: The Systematic Side Effects of Over-Prescribing Goal Setting". In: *Social Science Research Network* (2009).

[Pau+12] M. Paulitsch et al. "Evidence-based security in aerospace". In: *ISSRE Workshop 2012* (2012).

[PBD05] Klaus Pohl, Günter Böckle, and Frank J. van Der Linden. *Software Product Line Engineering - Foundations, Principles and Techniques*. Springer Science & Business Media, 2005.

[Pel+11] Jan Peleska et al. *Turn Indicator Model Overview*. Tech. rep. Department of Mathematics and Computer Science, University of Bremen, Germany, 2011.

[Per+12] Marie-Agnès Peraldi-Frati et al. "The TIMMO-2-USE project: Time modeling and analysis to use". In: *ERTS2 2012 - 6th International Congres on Embedded Real Time Software and Systems*. Toulouse, France, 2012, pp. 1–10.

[Poh+18] Richard Pohl et al. "Variant Management Solution for Large Scale Software Product Lines". In: *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*. New York, NY, USA, 2018, pp. 85–94.

[Poh10] Klaus Pohl. *Requirements Engineering - Fundamentals, Principles, and Techniques*. Springer-Verlag Berlin Heidelberg, 2010.

[Pol+12] Andreas Polzer et al. "Managing complexity and variability of a model-based embedded software product line". In: *Innovations in Systems and Software Engineering* (2012), pp. 35–49.

[PZ07] Malte L. Peters and Stephan Zelewski. "TOPSIS als Technik zur Effizienzanalyse". In: *WiSt - Wirtschaftswissenschaftliches Studium* (2007), pp. 9–15.

[RBC05] Meghan Revelle, Tiffany Broadbent, and David Coppit. "Understanding Concerns in Software: Insights Gained from Two Case Studies". In: *13th International Workshop on Program Comprehension*. St. Louis, USA, 2005, pp. 23–32.

[Rei04] Mark-Oliver Reiser. *Managing complex variability in automotive software product lines with subscoping and configuration links*. 2004.

[Ren+04] Xiaoxia Ren et al. "Chianti: A Tool for Change Impact Analysis of Java Programs". In: *Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*. New York, USA, 2004, pp. 432–448.

[Rin15] Tim Ring. "Connected cars - the next target for hackers". In: *Network Security* (2015).

[RM90] Harold E. Roland and Brian Moriarty. *System Safety Engineering and Management*. Canada: John Wiley and Sons, Inc., 1990.

[RQS12] Christine Rupp, Stefan Queins, and die SOPHISTen. *UML 2 glasklar: Praxiswissen für die UML-Modellierung*. Carl Hanser Verlag GmbH & Co. KG, 2012.

[RT01] Barbara G. Ryder and Frank Tip. "Change Impact Analysis for Object-oriented Programs". In: *Proceedings of the 2001 ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*. New York, USA, 2001, pp. 46–53.

[Saa02] Thomas L. Saaty. "Decision-making with the AHP: Why is the principal eigenvector necessary". In: *European Journal of Operational Research 145* (2002), pp. 85–91.

[Saa04] Thomas L. Saaty. "Decision making - the Analytic Hierarchy and Network Processes (AHP/ANP)". In: *Journal of Systems Science and Systems Engineering* (2004).

[Saa15] Christian Saad. "Data-flow based Model Analysis: Approach, Implementation and Applications". doctoralthesis. Universität Augsburg, 2015.

[SAG18] Siemens AG. *Warum Safety? Maschinensicherheit ist ein Muss!* `https://www.industry.siemens.com/topics/global/de/safety-integrated/maschinensicherheit/warum-safety/seiten/default.aspx`. accessed on September 13th, 2018. 2018.

[SAP19] SAP. *Change-Request-Management*. `https://help.sap.com/doc/saphelp_sm72_sp02/7.2.02/de-DE/4c/3acb82b50843b4e10000000a42189e/content.htm?no_cache=true`. accessed on March 21st, 2019. 2019.

[SC01] Nary Subramanan and Lawrence Chung. "Metrics for Software Adaptability". In: *Proceedings of The British Computer Society QualitySpecial Interest group's 9th Annual International Conference Software Quality Management (SQM 2001)*. 2001, pp. 95–108.

[Sch15] Fred Schenkelberg. *A Brief Introduction to Fault Tree Analysis*. https://accendoreliability.com/brief-introduction-fault-tree-analysis/. accessed on September 3rd, 2018. 2015.

[Sch18] Markus Schnappinger. "Interactive Visual Support for Exploration of Design Alternatives". MA thesis. Augsburg, Germany: University of Augsburg, 2018.

[Sch99] Eckehard Schnieder. *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme*. Vieweg Verlagsgesellschaft, 1999.

[Sei99] Ute Seiderer. *Panta rhei. Der Fluß und seine Bilder. Ein kulturgeschichtliches Lesebuch.* Reclam, 1999.

[SEJ08] Teodor Sommestad, Mathias Ekstedt, and Pontus Johnson. "Combining Defense Graphs and Enterprise Architecture Models for Security Analysis". In: *2008 12th International IEEE Enterprise Distributed Object Computing Conference*. 2008, pp. 349–355.

[SES15] SESAMO. *SESAMO Project Homepage*. http://sesamo-project.eu/. accessed on November 13th, 2017. 2015.

[Sho14] Adam Shostack. *Threat Modeling: Designing for Security*. Wiley Publishing, 2014.

[SIN03] Hirokazu Shimizu, Toshiyuki Imagawa, and Hiroshi Noguchi. "Reliability Problem Prevention Method for Automotive Components - Development of GD^3 Activity and DRBFM (Design Review Based on Failure Mode)". In: *International Body Engineering Conference & Exposition* (2003).

[SK90] Thomas L. Saaty and Joseph M. Katz. "How to make a decision: The Analytic Hierarchy Process". In: *European Journal of Operational Research*. Pittsburgh, USA, 1990, pp. 9–26.

[Som11] Ian Sommerville. *Software Engineering*. Pearson Studium - IT, 2011.

[Som18] Ian Sommerville. *Software Engineering 10th Edition*. https://iansommerville.com/software-engineering-book/web/critical-systems/. 2018.

[SP14] *SAFE Project Homepage*. http://www.safe-project.eu/. accessed on June 25th, 2015. 2014.

[SP16a] SafeCer. *SafeCer project homepage*. http://safecer.eu. accessed on November 13th, 2017. 2015.

[SP16b] SYNOPSIS. *SYNOPSIS project homepage*. http://www.es.mdh.se/SYNOPSIS/. accessed on June 25th, 2019. 2016.

[Spr12] John Spriggs. *GSN - The Goal Structuring Notation*. Springer-Verlag Berlin-Heidelberg, 2012.

[SR90] Dorothy E. Setliff and Rob A. Rutenbar. "Software Reusability". In: *Automatic Programming Applied to VLSI CAD Software: A Case Study*. Springer US, 1990, pp. 33–42.

[ST16] Andreas Sczepansky and Jürgen Triep. *An Introduction to Safety Critical Systems*. Tech. rep. QA Systems GmbH, 2016.

[Sta+11] R. Stahlmann et al. "STARTING EUROPEAN FIELD TESTS FOR CAR-2-X COMMUNICATION: THE DRIVE C2X FRAMEWORK". In: *Proceedings of 18th ITS World Congress and Exhibition 2011*. Orlando, Florida, USA, 2011.

[Tah+14] Ahmed Taha et al. "AHP-Based Quantitative Approach for Assessing and Comparing Cloud Security". In: *13th International Conference on Trust, Security and Privacy in Computing and Communications*. Darmstadt, Germany, 2014.

[TH19] *Techopedia Homepage*. `https://www.techopedia.com/definition/9269/scalability`. accessed on August 2nd, 2019. 2019.

[Tri00] Evangelos Triantaphyllou. *Multi-criteria Decision Making Methods: A Comparative Study*. Springer Science+Business Media B.V., 2000.

[WC11] Hongde Wang and Tiejun Cui. "Safety Assessment on railway crossings based on Extension AHP and Set Pairs Analysis". In: *International Conference on Management and Service Science (MASS)*. Dalian, China, 2011.

[Yu15] Yijun Yu. *Germanwings flight 4U9525: a victim of the deadlock between safety and security demands*. `http://theconversation.com/germanwings-flight-4u9525-a-victim-of-the-deadlock-between-safety-and-security-demands-39386`. accessed on August 30th, 2018. 2015.

[ZFW13] Youhu Zhao, Guicui Fu, and Bo Wan. "An Improved Risk Priority Number Method Based on AHP, Reliability and Maintainability Symposium (RAMS)". In: *2013 Proceedings - Annual*. Beijing, China, 2013.

List of Acronyms

AADL	Architecture Analysis & Design Language
ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ADT	Attack and Defence Tree
ADTA	Attack and Defence Tree Analysis
AES	Advanced Encryption Standard
AHP	Analytic Hierarchy Process
ATM	Automated Teller Machine
BbW	Brake-by-Wire
BC	Best Case
CAN	Controller Area Network
CEO	Chief Executive Officer
ChIA	Change Impact Algorithmic
CC	Cruise Control
CM	Counter Measure
ADAS	Advanced Driver Assistance System
DNF	Disjunctive Normal Form
DOS	Denial of Service
DRBFM	Design Review Based on Failure Mode
EAM	Enterprise Architecture Model
EBA	Emergency Brake Assist
ECU	Electronic Control Unit
EID	Extended Influence Diagram
E/E/PE	Electric, Elctronic and Programmable Electronic Systems
EMF	Eclipse Modeling Framework

FM	Feature Model
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects and Critically Analysis
FMEDA	Failure Mode, Effects and Diagnostic Analysis
FTA	Fault Tree Analysis
GPS	Global Positioning System
GSN	Goal Structuring Notation
HAZOP	Hazard and Operability Study
HP	Horse Power
KPI	Key Performance Indicator
LA	Lane Assist
LDP	Lane Departure Prevention
LIN	Local Interconnect Network
MC	Multi-Concerns
MCDM	Multi-Criteria Decision Making
OSD	Occurrence, Severity and Detection
OMG	Object Management Group
PCM	Pairwise Comparison Mode
PhD	Doctor of Philosophy
PIN	Personal Identity Number
PFMECA	Process Failure Mode Effect and Criticality Analysis
PLE	Product Line Engineering
POV	Point of Vulnerability
RCM	RPN Comparison Mode
ROA	Return on Attack
ROI	Return on Invest
RPN	Risk Priority Number

SGH	Safety Goal Hierarchy
SIL	Safety Integrity Level
SPL	Software Product Line
SST	Safety, Security and Timing
SSTM	Safety-, Security-, and Timing Model
SC	Safety Case
SCS	Safety-Critical System
SM	System Model
SMDS	Software Methodologies for Distributed Systems
SMS	Short Message Service
SysML	Systems Modeling Language
SuD	System under Development
TI	Turn Indicator
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TÜV	Technischer Überwachungsverein
UA	Utility Analysis
UML	Unified Modeling Language
WC	Worst Case
WCET	Worst Case Execution Time

List of Definitions

List of Figures

List of Tables

List of Algorithms