SAHARA

A Structured Approach for Hazard Analysis and Risk Assessments

Foto: Katharina Wieland Müller / pixelio.de

Sören Kemmann

SAHARA

A Structured Approach for Hazard Analysis and Risk Assessments

Vom Fachbereich Informatik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation
von

Dipl.-Inf. Sören Kemmann

Fraunhofer-Institut für Experimentelles Software Engineering
(Fraunhofer IESE)
Kaiserslautern

Berichterstatter:	Prof. Dr. Peter Liggesmeyer Prof. Dr. Norbert Wehn
Dekan:	Prof. Dr. Klaus Schneider
Tag der wissenschaftlichen Aussprache:	27.02.2015

D 386

Abstract

In this thesis, an approach is presented that turns the currently unstructured process of automotive hazard analysis and risk assessments (HRA), which relies on creativity techniques, into a structured, model-based approach that makes the HRA results less dependent on experts' experience, more consistent, and gives them higher quality. The challenge can be subdivided into two steps. The first step is to improve the HRA as it is performed in current practice. The second step is to go beyond the current practice and consider not only single service failures as relevant hazards, but also multiple service failures.

For the first step, the most important aspect is to formalize the operational situation of the system and to determine its likelihood. Current approaches use natural-language textual descriptions, which makes it hard to ensure consistency and increase efficiency through reuse. Furthermore, due to ambiguity in natural language, it is difficult to ensure consistent likelihood estimates for situations.

The main aspect of the second step is that considering multiple service failures as hazards implies that one needs to analyze an exponential number of hazards. Due to the fact that hazard assessments are currently done purely manually, considering multiple service failures is not possible. The only way to approach this challenge is to formalize the HRA and make extensive use of automation support.

In SAHARA we handle these challenges by first introducing a model-based representation of an HRA with GOBI. Based on this, we formalized the representation of operational situations and their likelihood assessment in OASIS and HEAT, respectively. We show that more consistent situation assessments are possible and that situations (including their likelihood) can be efficiently reused. The second aspect, coping with multiple service failures, is addressed in ARID. We show that using our tool-supported HRA approach, 100% coverage of all possible hazards (including multiple service failures) can be achieved by relying on very limited manual effort. We furthermore show that not considering multiple service failures results in insufficient safety goals.

Table of Contents

List of Figures

List of Tables

1 Introduction

In this thesis, we will use Figure 1.1 as a recurring guideline throughout the thesis and indicate the location in the figure at the beginning of the corresponding section.

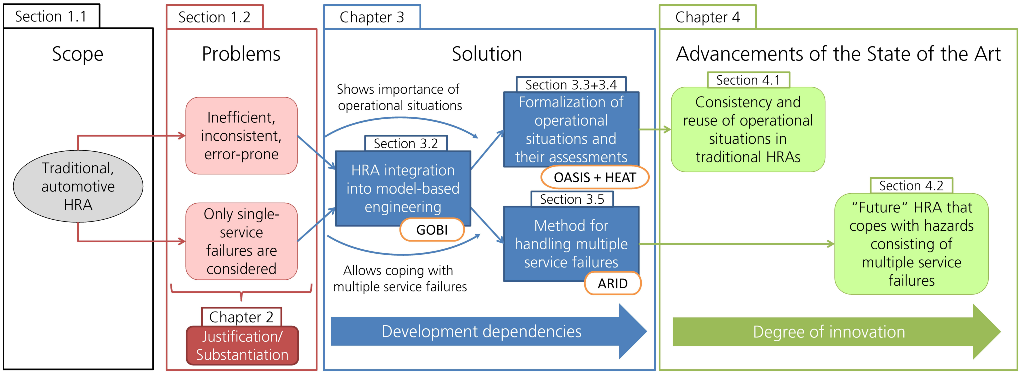

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method for handling multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 1.1: Schemantic overview of this thesis

In the subsequent sections of this chapter, we will first give a short introduction to the scope of this thesis and then introduce the problems we are addressing in this thesis.

1.1 Motivation

The evolution of embedded systems has shown one immanent trend: Electric, electronic, and programmable electric systems are increasingly becoming the driver for innovations and an enabler of new features and functionalities. In 2015, the cost of electronics in road vehicles will amount to over 40% of the overall costs [fAR11]. The state of the practice, especially in automotive systems, is far away from single independent subsystems, but the E/E system is characterized by complex networks of interacting subsystems. Luxury class vehicles contain more than 80 electronic control units (ECUs) [Ber03]. Nowadays, many ECUs communicate with each other, and the tendency is for this to increase even more in the future.

Safety has always been one of the most important properties in such systems. The ever increasing complexity of modern systems does however impose new challenges when it comes to ensuring the safety of a system. The process for making a system safe can be divided into two phases: (1) safety requirements elicitation and (2) implementation of these requirements. It is common knowledge that precise, complete, and correct requirements are the key to success. Leveson pointed this out already in 1995[Lev00]: "The majority of software-related accidents are caused by requirements errors." For safety engineering, this means that complete and correct safety requirements are indispensable.

While the equal importance of the elicitation and implementation of safety requirements is common knowledge, this balance is not reflected in the maturity of the respective safety engineering phases. The implementation of safety requirements uses structured methods and state-of-the-art techniques such as formal models and model-based development. In contrast, the elicitation of safety requirements is often performed unsystematically and is, to a large extent, based on creativity techniques.

In the automotive domain, the process for eliciting top-level safety requirements is termed hazard analysis and risk assessment (HRA). In this thesis, we advance the state of the art by providing a model-based, structured approach for HRAs.

1.2 Problem Statement

In the previous section, we pointed out that the maturity of hazard analysis and risk assessments lags far behind the maturity of safety requirements implementation. The goal of this thesis is to push HRAs to a more mature level by filling the gaps currently existing in the state of the art of HRAs. In this section, we present an overview of these gaps and the respective challenges that need to be tackled on the way to a structured approach for hazard analysis and risk assessments.

While the overall goal and general challenge is to find an integrated model-based approach, this challenge can be partitioned into sub-challenges. As the term implies, the task of HRA can be split into two parts. (1) the hazard analysis and (2) the risk assessment. The first part deals with the task of deciding which critical concomitances of hazards and operational situations might occur, while the second part quantifies the risk of such a concomitance. This is schematically shown in Figure 1.2. The oval shapes represent tasks, the rectangular shapes represent the resulting artifacts.

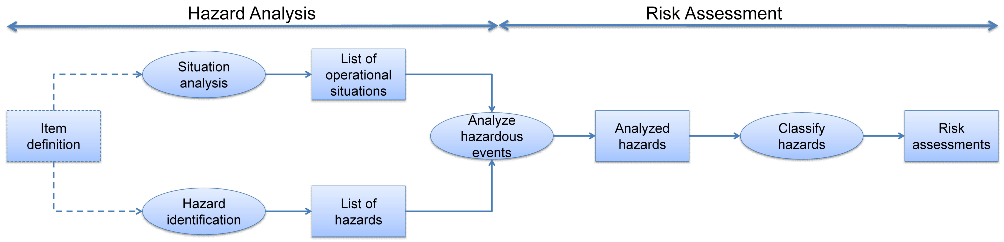


Figure 1.2: Hazard Analysis and Risk Assessment process

1.2.1 Problem discussion of the hazard analysis part

In the ***hazard analysis*** part, three tasks need to be performed. The starting point is usually the hazard identification. It deals with the elicitation of possible system-level failures and corresponding hazards. In parallel, the safety engineer has to prepare a list of operational situations that express relevant environmental conditions and operating modes that may allow a hazard to transition to a harmful event, i.e., an accident. The hazardous event analysis deals with this harmful event by "pairing" hazards with situations and reasoning about the effect.

The basic challenge of a hazard identification is completeness. If one omits or forgets a hazard the possibility exists that a critical concomitance might not be considered. The situation analysis has to be correct in the sense that an operational situation should lead to harm, if the respective hazard occurs. A hazard describes a misbehavior at system-level. For being a valid operational situation (for the misbehavior), the effect of the misbehavior in the operational environment must result in a harmful event. If the situation is to general and a possibility exists that the misbehavior is not harmful, the situation is not valid. The combination of an operational situation and a hazard (the misbehavior) is termed hazardous event. The analysis of hazardous events is only a matter of double checking whether each situation on the list is valid with respect to the hazard. In practice this is often considered a challenge, but it is only challenging because the validity of each situation is often checked in this step not in the step of situation analysis. Thus, if we assume that the situations are correctly analyzed in the situation analysis step, the analysis of hazardous events is not necessary anymore and therefore does not impose a scientific challenge anymore. We therefore do not discuss this aspect any further.

The *completeness requirement of hazard identifications* addresses essentially the aspect of eliciting all system-level service failures. A system consists of multiple services and each service usually has multiple possible failures. A hazard is a subset of possible service failures. The complete set of hazards is therefore the power set of the set of single service failures. Completeness can therefore be subdivided into coming up with a complete set of single service failures and building the power set. The identification or definition of single service failures is already quite mature, thanks to existing hazard identification methods. The creation of a power set is trivial. In contrast, the *assessment* of this power set is a huge challenge. This will be discussed as part of the risk assessment section.

The *correctness requirement of a situation analysis* implies two fundamental challenges.
(1) The challenge of formalizing operational situations in general and the environment specifically:
Due to unlimited influence factors in the environment, the number of possible situations is unlimited as well. The challenge of formalizing situations means essentially that one needs to handle the potentially unlimited set of influence factors on the one hand and that one needs a way of representing operational situations on the other hand. In practice situation are defined mostly textual and contained in lists. However, this is not sufficient because it is not possible to decide whether a situation is valid to be used in combination with a respective hazard. If the situation for example contains "ice", it is not clear which aspect the safety engineer wanted to express? Was it the information that it is cold outside, or that the underground might be slippery. Without this information the situation definition is not useful. Additionally, a list has the disadvantage that one needs to go through the complete list in order to collect *all* relevant situations and systematically identifying situations not on the list is difficult. This leads us to our second challenge.
(2) The challenge of reusing operational situations. Because situations are independent its possible failures/hazards, they are an obvious artifact for reuse. Reusing situations does, however, impose new challenges. Deciding which situations to reuse is not trivial. Situations need to be appropriate in

the sense that all sufficient but only necessary details are described. Lists do not support this. Systematic selection of appropriate situations based on lists is not possible. In practice, this results in situations being selected that are close to the real situation, but not optimal. Another disadvantage of a situation list is continued "learning" on lists. By just extending a list with every new situation ever encountered, one would end up with an enormous list that would not be usable anymore. Therefore, a more structured approach for defining, retaining, retrieving, and reusing situations is required.

1.2.2 Problem discussion of the risk assessment part

A similar picture can be sketched for the second part, the ***risk assessment***. The task is that the safety engineer takes the list of hazardous events and needs to process the list in order to come up with a risk assessment for each of these hazardous events. We subdivide this challenge into (1) coming up with a risk value for one hazardous event and (2) determining a risk assessment value for each of these hazardous events on the list.
(1) In the automotive domain the criticality/risk of a hazardous event is expressed in terms of an integrity level (ASIL[1]). The determination of the ASIL value is in ISO 26262 defined by three parameters:

1. *exposure*, describing the probability of occurrence of the operational situation;
2. *severity*, defining the degree of harm of a possible accident;
3. *controllability*, which describes the likelihood that a person, usually the driver, can prevent the occurrence of harm.

We subdivide the classification challenge into the determination of the values of the three parameters:
The assessment of *severity* requires in-depth knowledge in medicine in order to support specific reasoning about possible injuries. The formalization of the determination of the severity value is therefore out of scope of this thesis.
The same is true for the *controllability* parameter. To reason about controllability, profound knowledge in human behavior modeling and human-machine interaction mechanisms is necessary. Even though we define what the core information from a human-machine interaction model is and include this information as interface in our model, we do not define a concrete controllability model in this thesis.
In contrast to the other parameters, the *exposure* parameter is fully included in this thesis. Formalizing operational situations was already one of the major challenges in the first part, thus it is straightforward that the assessment of the exposure parameter should be tightly coupled with the situation model. If we assume that we have a structured way of representing situations, the challenge remains of how to consistently assess the exposure

[1] ASIL = Automotive Safety Integrity Level

parameter. In practice, this is done (for the limited situation lists) in laborious cross-OEM/tier meetings in order to find a common classification. This is, however, very time-consuming, error-prone, and works only for very limited lists. Additionally, there is no way to decide whether the situations are consistently assessed amongst each other: A situation that is a specialization of another situation should obviously be less (or equally) probable as a more general situation. Thus, to finalize the model-based consideration of operational situations, the challenge remains to include the exposure parameter assessment in the model construction as well and to support automated reasoning about situations already assessed before.

(2) The second challenge is to process the complete list of hazardous events. In practice this is not considered a challenge, because only hazards consisting of single service failures are listed. A bigger challenge arises if we assume complex network of functions as described in the introduction. There is obviously a strong interdependence between functions and we therefore have to accept that one internal failure may result in multiple simultaneous service failures on the system level. As an example of this, ISO 26262 [ISO11] brings up the power supply: If the power supply is lost, multiple systems will be affected at the same time: The power steering, the head lights, and even the engine torque are all lost at once. Thus, the list of hazardous events should obviously include hazards consisting of more than one service failure. However, if we consider a car with 100 system-level services and only one failure mode for each service, in a worst-case scenario we would end up analyzing $2^{100} = 10^{30}$ hazards. Thus, the challenge is to find a way to argue the completeness of hazards without actually assessing all hazards. By assessing all hazards we mean the manual assessment task a safety engineer has to perform.

1.2.3 Summary of problem statement

Figure 1.3 shows an overview of the problem statement identified in this section.

We summarize the major challenges for finding a structured approach for HRAs as follows:

1. To formalize operational situations, including their exposure assessment, in such a way that enables a safety engineer to ensure situation consistency and support the systematic reuse of situations.
2. To handle the enormous number of system-level hazards when moving from single service failures to multiple (simultaneous) service failures.

Furthermore, we acknowledge that the assessment of the severity and controllability parameters in a more structured way is an open topic. These topics should, however, be addressed by experts from the medical and human behavior domains. We do not address these challenges in this thesis.

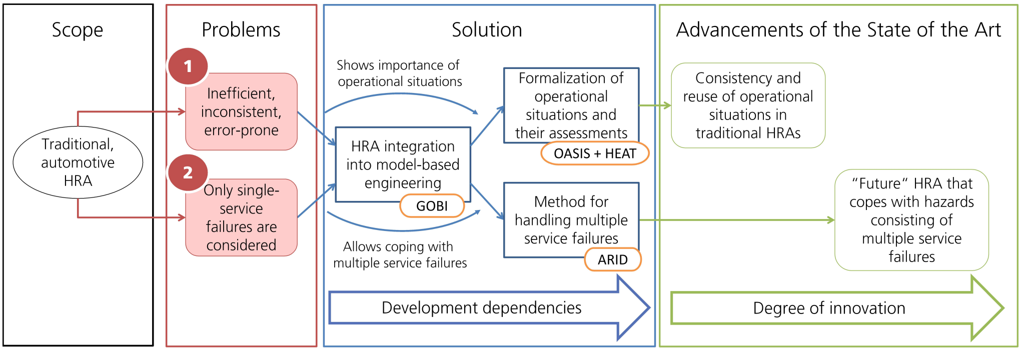

Scope
Problems
Solution
Advancements of the State of the Art
Traditional, automotive HRA
1
Inefficient, inconsistent, error-prone
2
Only single-service failures are considered
Shows importance of operational situations
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Formalization of operational situations and their assessments
OASIS + HEAT
Method for handling multiple service failures
ARID
Development dependencies
Consistency and reuse of operational situations in traditional HRAs
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 1.3: PhD overview including major challenges/problems

2 State of the Art and Related Work

In this chapter, we further structure the challenges arising when we aim at complete and correct safety goals. Our work is based on the HRA process from the automotive domain and therefore aligns with ISO 26262 [ISO11]. Hence, this chapter is structured as follows. We begin by introducing the ISO 26262 process of an HRA. This especially includes clarification of the terms used in safety engineering. Afterwards we will discuss the state of the art. As a result of this discussion, we will provide concrete challenges. In the last part, we take these challenges and point out related work that can help us to handle these challenges.

The goal of this chapter is to give justification and substantiation to the problems identified in the problem statement. Figure 2.1 shows this as an overview.

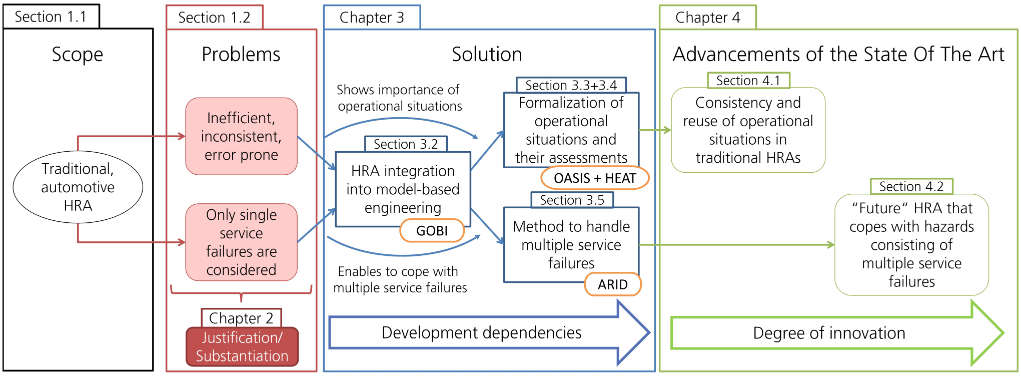

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error prone
Only single service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Enables to cope with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method to handle multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State Of The Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 2.1: Logical localization of the state-of-the-art chapter

2.1 Hazard Analysis and Risk Assessment Fundamentals

Because the central topic of this thesis is HRA, we introduce ISO 26262's process of an HRA here. This has two purposes. First, this will present the basic terms and practices and will show their usage in HRA. Second, every domain has a slightly different concept of HRA. Hence, in order to start with a common basis, it is necessary to point out the concrete process as described in ISO 26262.

"The objective of the hazard analysis and risk assessment is to identify and categorize the hazards of the item and formulate the safety goals related to the prevention or mitigation of these hazards, in order to avoid unreasonable risk"[ISO11]

2.1.1 Important terms

In order to fully understand this quote, we need to discuss what is meant by the terms: "*hazard*", "*item*", "*safety goals*", and "*risk*". Some of these terms were informally introduced in the motivation, but we will discuss them in more detail from ISO 26262's point of view in this section.

The standard talks about "*hazards of the item*"; furthermore, the first requirement in the HRA section states that the HRA "*shall be based on the item definition*". In ISO 26262 the term "item" refers to a "system or array of systems or a function to which ISO 26262 is applied". The item definition itself has two objectives:

- The first objective of the item definition is to define and describe the item.
- The second objective is to support an adequate understanding of the item so that each activity defined in the safety life-cycle can be performed.

Thus, the item definition requires a profound understanding of the item under development. If the item is not the whole system, "*it shall be ensured that the boundary of the item and the item's interface, as well as assumptions concerning other items and elements, are determined* [...]"[5.4.3]. This implies that one cannot slice the system into single simple functions and perform an HRA on these functions without considering other items or elements. This requirement establishes the demand for analyzing the complex network of functions instead of single functions.

In ISO 26262, a "*hazard*" is defined as "*potential source of harm*". The definition itself is very fuzzy, because every factor contributing to an accident could be considered as a "*source of harm*". This definition, however, is the common definition found in various standards and in the literature. Besides this definition, one usually finds a hint on how to interpret this definition in the concrete setting. Therefore, it is not surprising that interpretations range from being a "*state of a system and its environment* [...]"[Lig09] to being "*any biological, chemical, mechanical, or physical agent that is reasonably* *likely to cause harm or damage to humans, other organisms, or the environment in the absence of its control*" [wik13]. In section 7.4.4of ISO 26262: "*Situation analysis and hazard identification*", the term "*hazard*" is further refined: "*Hazards shall be defined in terms of the conditions and events that can be observed at the vehicle level.*" Thus, ISO 26262 excludes the environment from the definition. The term that includes the environment, as in the definition of Liggesmeyer [Lig09], is "*hazardous event*". Because this thesis is based on the view of ISO 26262, we will use ISO 26262's wording of separating hazards from hazardous events.

Closely related to the term hazard is the term risk. The relation is that risk gives us a comparative score for classifying hazards. In ISO 26262, it is defined as:

"Risk = combination of the probability of occurrence of harm and the severity of that harm."

Annex B of ISO 26262 part 3 gives a more detailed picture of "*risk*". It says that "a risk R can be described as a function F, with the frequency f of occurrence of a hazardous event, the ability of the avoidance of specific harm or damage through timely reactions of the persons involved (C = controllability), and the potential severity of the resulting harm or damage (S = severity):

$$R = F(f, C, S) \tag{2.1}$$

The frequency f is, in turn, influenced by two factors:

- One factor to consider is how frequently and for how long individuals find themselves in a situation where the aforementioned hazardous event can occur. In ISO 26262, this is simplified as being a measure of the probability of the driving scenario taking place in which the hazardous event can occur (E = exposure).
- Another factor is the failure rate of the item that could lead to the hazardous event (lambda = failure rate). This factor is characterized by undetected random hardware failures and by hazardous systematic faults remaining in the system.

$$F = E \times \lambda \tag{2.2}$$

Because development in accordance with ISO 26262 leads to safe systems, the resulting ASIL determines the minimal set of requirements for compliance with the item, in order to avoid random hardware failures and systematic faults. For this reason, the lambda of the item is not considered in the risk assessment.

Summarizing, we can state that:

$$\textbf{\textit{Risk}} = \textbf{\textit{Probability of Occurrence of Harm}} \times \textbf{\textit{Severity of Harm}} \tag{2.3}$$

The "*probability of occurrence of harm*" can furthermore be refined as a combination of the failure rate lambda, the exposure E, and the controllability C. Thus:

$$Risk = (\lambda \times E \times C) \times S \tag{2.4}$$

Developing a system according to ISO 26262 leads to a safe system and implies that the residual risk is acceptable:

$$Risk_{Acceptable} = (E \times C) \times S \tag{2.5}$$

The missing parameter "lambda" is encoded in the required ASIL. The higher lambda, the more frequent failures would occur and the more effort has to be spent on making the system safe, thus the higher the ASIL.
It is important to note that the ASIL is fully determined by the three parameters Severity S, Exposure E, and Controllability C.

Severity:
"Estimate of the extent of harm to one or more individuals that can occur in a potentially hazardous situation" [ISO11]

Table 2.1 shows the severity classes as defined in Part 3 of ISO 26262.

Table 2.1: Severity classes
(Table 1 of Part 3 of ISO 26262, [ISO11])

	Class			
	S0	**S1**	**S2**	**S3**
Description	No injuries	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries

In addition to the normative table 2.1, there is a more concrete table listed in the informative annex B of part 3. Table 2.2 shows this table.

Exposure:
"State of being in an operational situation that can be hazardous if coincident with the failure mode under analysis" [ISO11]

Table 2.3 shows the classes of probability of exposure regarding operational situations as defined in Part 3 of ISO 26262.

In addition to the normative table 2.3, there is a more concrete table listed in the informative annex B of part 3. Table 2.4 shows this table.

Controllability:
"Ability to avoid a specified harm or damage through the timely reaction of the persons involved, possibly with support from external measures" [ISO11]

Table 2.2: Examples of severity classification (Table B.1 in Part 3 of ISO 26262, [ISO11])

	Examples of severity classification based on AIS [HJE+10]			
	S0	**S1**	**S2**	**S3**
Reference for single injuries (from AIS scale)	AIS 0 - Damage that cannot be classified as safety-related, e.g. bumps with roadside infrastructure	More than 10% probability of AIS 1-6 (and not S2 or S3)	More than 10% probability of AIS 3-6 (and not S3)	More than 10% probability of AIS 5-6

Table 2.3: Classes of probability of exposure regarding operational situations (Table 2 of Part 3 of ISO 26262, [ISO11])

	Class				
	E0	**E1**	**E2**	**E3**	**E4**
Description	Incredible	Very low probability	Low probability	Medium probability	High probability

Table 2.4: Classes of probability of exposure regarding duration in operational situations (Table B.2 in Part 3 of ISO 26262, [ISO11])

	Class of probability of exposure			
	E1	**E2**	**E3**	**E4**
Duration (% of average operating time)	Not specified	<1% of average operating time	1% to 10% of average operating time	>10% of average operating time

Table 2.5 shows the classes of controllability as defined in Part 3 of ISO 26262.

Table 2.5: Controllability classes (Table 3 of Part 3 of ISO 26262, [ISO11])

	Class			
	C0	**C1**	**C2**	**C3**
Description	Controllable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable

In addition to the normative table 2.5, there is a more concrete table listed in the informative annex B of part 3. Table 2.6 shows this table.

The result of an HRA is a safety goal. A safety goal is a "top-level safety requirement as a result of the hazard analysis and risk assessment." Assigned

Table 2.6: Examples of hazardous events possibly controllable by the driver of by the persons potentially at risk
(Table B.4 in Part 3 of ISO 26262, [ISO11])

	Controllability classes			
	C0	**C1**	**C2**	**C3**
Description	Controllable in general	99% or more of all drivers or other traffic participants are usually able to avoid harm	90% or more of all drivers or other traffic participants are usually able to avoid harm	Less than 90% of all drivers or other traffic participants are usually able, or barely able, to avoid harm

to each safety goal is an ASIL. This gives the safety goal the information of how stringent this safety goal needs to be achieved in order to end at an acceptable level of risk.

2.1.2 ISO 26262's Hazard Analysis and Risk Assessment process

Following the introduction of the terminological basis, we will now present how to perform an HRA. Figure 2.2 shows an overview of the process. The oval elements display activities while the rectangular boxes represent artifacts.

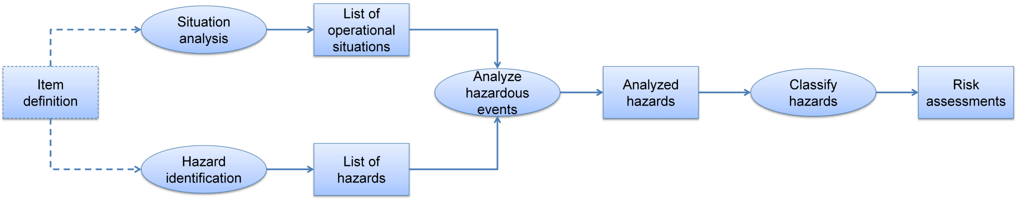


Figure 2.2: ISO 26262's Hazard Analysis and Risk Assessment process

The item definition, i.e., the information about the system or function the HRA is scoped to, is the general input to every HRA. The process starts from the left with two activities: the situation analysis and the hazard identification. In the situation analysis, one shall prepare a list of "*operational situations and operating modes in which an item's malfunctioning behavior is able to trigger hazards*". Another constraint for the preparation of the list of situations is given by requirement 7.4.7: "*During hazard analysis and risk assessment, having established the list of operational situations* [. . .], *it shall be ensured that the chosen level of detail of the list of operational situations does not lead to an inappropriate lowering of the ASIL of the corresponding safety goals.*"

This requirement seems to be relevant for the determination of the ASIL value in the classification step. This is true in the sense that the effect will be evident there, the challenge is, however, a challenge of the situation analysis. shows its relevance later in the classification step, but should be considered in the situation analysis. An operational situation should describe sufficient conditions that a hazard transitions to an accident, but should only describe necessary conditions. If the operational situation is split into small sub-situations, the splitting factor is not necessary for the hazard to transition to an accident and therefore lowers the probability of the situation inappropriately.

The other activity, hazard identification, is related to the system or the function under consideration. Here, a list of "*hazards of the item shall be determined systematically*". This relates to the possible failures of the system observable at the vehicle level. The next step combines both lists. Each hazard is analyzed for possible consequences if combined with each situation; thus, hazardous events are built. In this step, "*consequences of hazardous events shall be identified* [...]". After this step, the whole hazardous scenario is built: The system failure, the resulting (mis-)behavior, the influence on and of the environment, and the expected harm. The last step quantifies the risk by classifying the scenario with the parameters introduced above: Exposure, Controllability, and Severity.

The result of this classification is that "*a safety goal shall be determined for each hazardous event evaluated in the hazard analysis*" and that "*the ASIL determined for the hazardous event shall be assigned to the corresponding safety goal.*"

2.2 Core Challenges and Goals

We structure our discussion of the state of the art and related work along our HRA process introduced in the introduction:

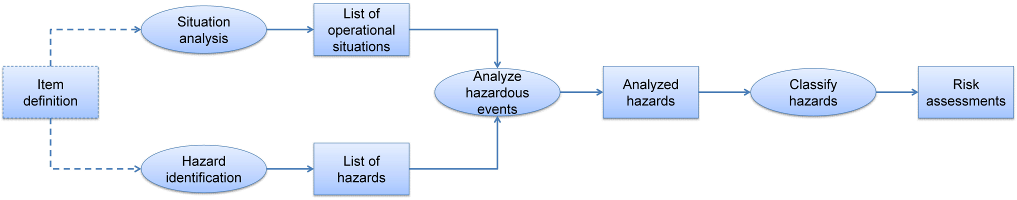


Figure 2.3: Hazard Analysis and Risk Assessment process

Our related work is therefore classified into the areas the work addresses in accordance with the problem statement:

⇛ ***Holistic model-based approach for a structured HRA process***

- ▶ Structuring of hazard analysis
  - ▷ Correctness of situation analysis
    - ⇛ Formalization of operational situations
    - ⇛ Efficient reuse of operational situations
  - ▷ Identification of system-level failures
- ▶ Structuring of risk assessments
  - ▷ Hazardous event classification
    - ⇛ Controllability assessment
    - ⇛ Severity assessment
    - ⇛ Exposure assessment
  - ▷ Dealing with multiple service failures

We will further classify the related work into two categories:

1. Work that directly addresses one of the challenges above, i.e, work that claims to have solved one or more of the above challenges.
2. Work that can be used to solve the problems, but was not developed for use in HRA.

In the next section, the state of the art is reviewed, which discusses mostly work of category one. Category two work will then be discussed in the related work section.

2.3 State Of The Art

The awareness that the existing methods and techniques for HRAs are not optimal and are still a field of research is evident in the fact that this topic is being explicitly included in current research project applications. In SPES 2020 XT, for example, an explicit request for model-based HRAs was input by industry. There is, however, related work dealing with this challenge already.

2.3.1 ATESST2

In the automotive domain itself, there have been some research activities in the area of functional safety. ***ATESST2*** [ATE13] stands for "Advancing Traffic Efficiency and Safety through Software Technology - Phase 2" and was a European research project based in FP7. The goal of ATESST 2 was to improve the management of complexity and to handle the criticality of modern vehicles. To this end, ATESST2 provided means for integrating the engineering information found in documents, spreadsheets, and legacy tools into one systematic structure. Thus, the goal was a model-based approach for system-model-integrated safety analysis. The project addressed the whole safety lifecycle. We will focus on the HRA part of this lifecycle and will discuss their findings. Deliverable 2.1 (D2.1) [WMP$^+$10] is named "Review of relevant Safety Analysis Techniques" and gives an overview of the state of the practice and the state of the art from the industry's point of view. For HRA, this review is limited to giving some guidelines on how to perform certain tasks from ISO 26262's HRA process. In section 4.4 of D2.1, a scenario definition is proposed. Figure 2.4 shows the influence factor classes they provided.

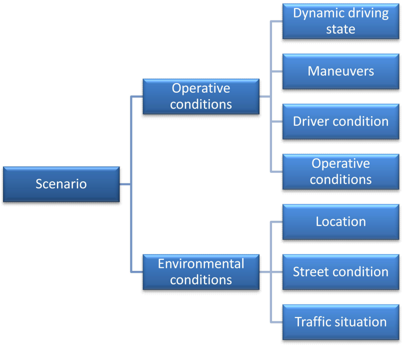


Figure 2.4: ATESST2's influence factor classes

This model is only a suggestion, however, and it is not meant to be complete. But it is a good refinement of the general term "situation" (in ISO 26262). In section 4.5 (of D2.1), hazard identification is addressed. It is stated that "all possible functional anomalies derivable from each foreseeable source, either internal (system faults) or (e.g. foreseeable misuse) shall be defined". Furthermore, an exhaustive list of anomalies of function activation is provided:

- Unwanted activation
- Missed activation
- Degraded activation (vs. value or timing)
- Incoherent activation (in the presence of more than one function's output)
- Unwanted deactivation
- Missed deactivation
- Degraded deactivation (vs. value or timing)
- Incoherent deactivation

This list is consistent with the HAZOP guideword approach [Kle92] and Papadopoulos's Functional Failure Analysis approach [PM99]. In addition to these functional failure classes, the notion of Hazard is refined in section 4.5.3 (of D2.1): "Hazard is a potential source of harm, due to an item's malfunction in concomitance with a particular scenario's condition. Therefore a malfunction can be cause of several hazards with different consequences, really as a function of the involved scenario." Additionally, the following figure is provided to make the relation between malfunction, scenario, and hazard clear:

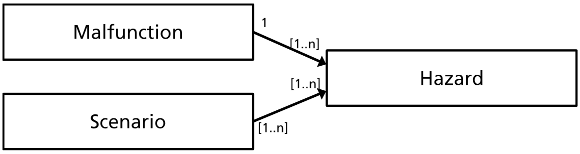


Figure 2.5: Relation between malfunction, scenario, and hazard

However, this figure is misleading if one has the ISO 26262 definitions of the respective terms in his mind: In ATESST2 a hazard describes with the concomitance of a malfunction and a scenario in terms of ISO 26262 a hazardous event. A malfunction would be one possible hazard. To be precise, it would be a single service failure. For the term "scenario" ISO 26262 would probably use "situation". Figure 2.6 shows the correct figure from ISO 26262 point of view. Please note that the besides the terms, the multiplicity located close to the "Situation/Scenario" block changed.

This inaccuracy of terms is particularly interesting due to the fact that some members of the ATESST2 consortium were involved in the development of ISO 26262 as well. For us, this reinforces our conviction that this topic has to be addressed in a more holistic and formalized way.
ATESST2 additionally propose to use the so called, Cooper-Harper scale for

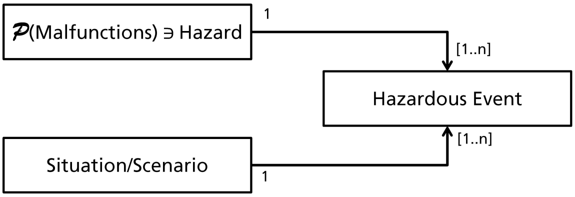


Figure 2.6: ISO 26262's view on hazard, scenario/situation, and hazardous event

a better controllability argumentation. The Cooper-Harper scale was suggested in 1969 for evaluating the aircraft response related to pilot actions. A pilot's performance or better the success of a pilot's action is rated on a scale from 1 to 10, where 1 is kind of "easily controllable" and 10 is "uncontrollable". The rating process is guided by three questions to help the engineer achieve a reasonable result. Because ISO 26262 has a five-level scale only, the ATESST2 group defined a mapping from the ten-level scale of the Cooper-Harper approach to the five-level ISO 26262 scale. Besides this interesting adaptation for controllability, the ATESST2 deliverable contains only limited additional information compared to ISO 26262's Annex B (Hazard analysis and risk assessment) of part 3.

We can conclude that the ATESST2 approach is very close to practice. Therefore, it especially provides valuable insights into the automotive domain and the state of the practice. As such, it addresses many challenges we identified. The solutions provided, however, are more practical guidelines than scientific advances. The environment formalization is a good first approach because it provides possible influencing factors of operational situations. The exhaustive list of system-level failure identification guidewords is a good domain-specific adaptation of the HAZOP and FFA approaches and can be used directly for hazard identification. The risk assessment does not provide any new information on assessing the ASIL parameters, with the exception of the review of the Cooper-Harper scale adaptation for controllability.

2.3.2 EASIS

Another big research project in the domain of automotive safety engineering was called "***Electronic Architecture and System Engineering for Integrated Safety Systems - EASIS***". The goal of the project was to provide an integrated view on vehicle development. This means system engineering principles were applied to the vehicle as a whole. We should point out that the EASIS project was completed in parallel with ISO 26262 development and was finished three years before the final publication of ISO 26262. Many EASIS members were involved in ISO 26262's development and therefore the results of the EASIS project should be treated as additional information to ISO 26262. In deliverable 3.2 part 1 "Guidelines for establishing dependability requirements and performing hazard analysis", our notion of separating system-internal and system-external analysis is supported by Figure 2.7.

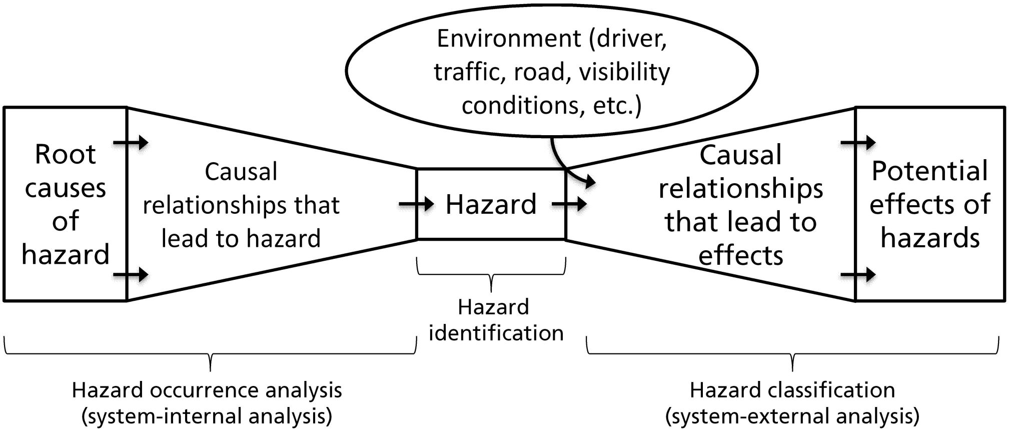

Environment (driver, traffic, road, visibility conditions, etc.)
Root causes of hazard
Causal relationships that lead to hazard
Hazard
Causal relationships that lead to effects
Potential effects of hazards
Hazard identification
Hazard occurrence analysis (system-internal analysis)
Hazard classification (system-external analysis)

Figure 2.7: Bow tie diagram for hazard analysis

Therefore EASIS supports ISO 26262's hazard definition by stating that:

"A hazard is an undesirable condition or state of a vehicle that could lead to an undesirable outcome depending on other factors that can influence the outcome."

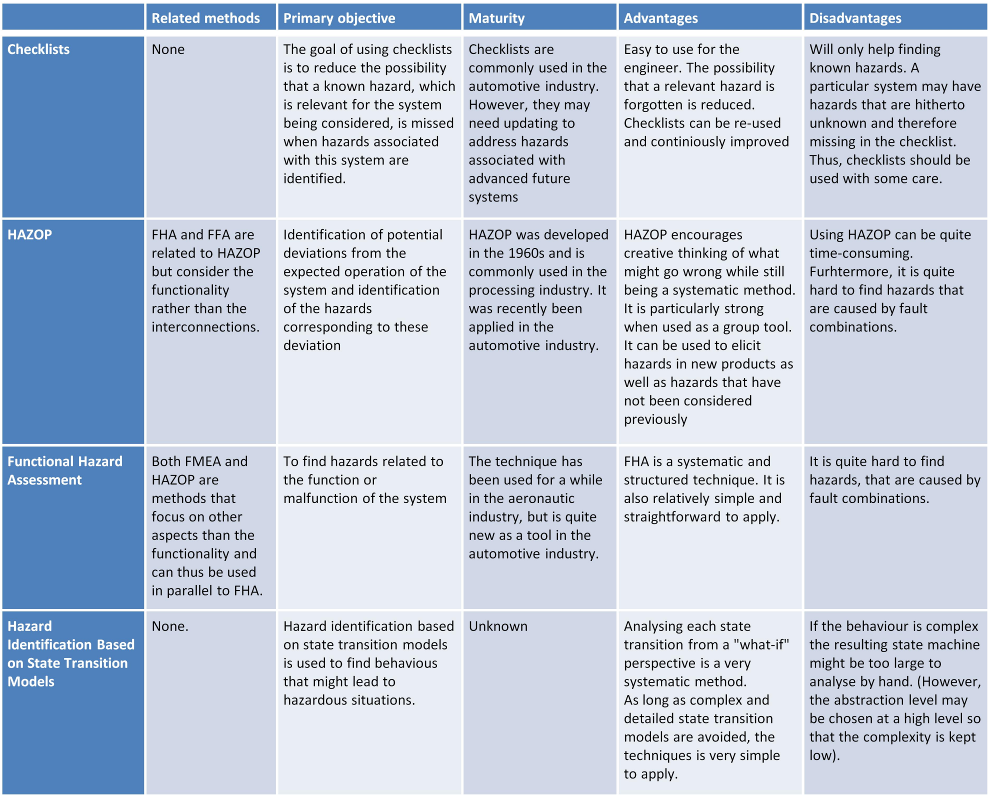
Based on this figure, three fundamentally different ways are described to identify a hazard.

1. Through a causal analysis of the system-external effects. The movement of the vehicle could, for example, deviate from the intended movement. Starting from this, one could try to find causes for the deviation of the movement. One example could be a "self-steering vehicle".
2. Through a forward analysis from system-internal causes to vehicle-level effects. If one considers, for example, that the ESC-ECU fails, one would ask what might happen. One effect could be the loss of stability control.
3. Through an analysis of the border between system as a function and the vehicle in the physical domain. The general starting point for such an analysis is the actuator-level.

In additional to this classification, an evaluation of hazard identification methods was performed. The techniques and the basic results are given in Table 2.7.

Based on the review of available hazard identification methods, a recommended approach was formulated: "The most effective way to identify hazards appears to be to use several methods since they complement each other. Checklists can be used for an easy start followed by HAZOP, FHA

Table 2.7: Hazard identification classification and assessment



	Related methods	Primary objective	Maturity	Advantages	Disadvantages
Checklists	None	The goal of using checklists is to reduce the possibility that a known hazard, which is relevant for the system being considered, is missed when hazards associated with this system are identified.	Checklists are commonly used in the automotive industry. However, they may need updating to address hazards associated with advanced future systems	Easy to use for the engineer. The possibility that a relevant hazard is forgotten is reduced. Checklists can be re-used and continiously improved	Will only help finding known hazards. A particular system may have hazards that are hitherto unknown and therefore missing in the checklist. Thus, checklists should be used with some care.
HAZOP	FHA and FFA are related to HAZOP but consider the functionality rather than the interconnections.	Identification of potential deviations from the expected operation of the system and identification of the hazards corresponding to these deviation	HAZOP was developed in the 1960s and is commonly used in the processing industry. It was recently been applied in the automotive industry.	HAZOP encourages creative thinking of what might go wrong while still being a systematic method. It is particularly strong when used as a group tool. It can be used to elicit hazards in new products as well as hazards that have not been considered previously	Using HAZOP can be quite time-consuming. Furhtermore, it is quite hard to find hazards that are caused by fault combinations.
Functional Hazard Assessment	Both FMEA and HAZOP are methods that focus on other aspects than the functionality and can thus be used in parallel to FHA.	To find hazards related to the function or malfunction of the system	The technique has been used for a while in the aeronautic industry, but is quite new as a tool in the automotive industry.	FHA is a systematic and structured technique. It is also relatively simple and straightforward to apply.	It is quite hard to find hazards, that are caused by fault combinations.
Hazard Identification Based on State Transition Models	None.	Hazard identification based on state transition models is used to find behavious that might lead to hazardous situations.	Unknown	Analysing each state transition from a "what-if" perspective is a very systematic method. As long as complex and detailed state transition models are avoided, the techniques is very simple to apply.	If the behaviour is complex the resulting state machine might be too large to analyse by hand. (However, the abstraction level may be chosen at a high level so that the complexity is kept low).

and 'Hazard Identification Based on State Transition Models' for the different aspects they bring into the analysis."
We support this idea of combining several techniques to cover different aspects. The piece missing to make this optimal would be a guideline on which approaches to combine in which situation. Furthermore, the precise prerequisites for applying the approaches are not 100% clear, especially for the state-based approach. Applicability is called a matter of abstraction, but guidance is neither given on how to achieve this abstraction nor on what a correct level of abstraction is.

The EASIS project does not explicitly propose or review a process for performing an HRA, but reviewed different hazard classification approaches and derived its own EASIS hazard classification method.
EASIS provides a very good review on hazard identification methods. The suggestion to combine more than one technique is reasonable. One deficiency we see in this approach is that final guidance on how to combine the various techniques is missing. The challenge of structuring the risk assessment is addressed directly. However, the result has the same maturity as the approach described in ISO 26262. Thus, while this work is probably the basis

for ISO 26262, it is a very good research result. Due to the fact that it was refined and specialized in ISO 26262 and our focus is on compliance with this standard, the fundamental work provided by EASIS does not advance our work.

2.3.3 EmMORI

The ***EmMORI*** approach by Ständer [Stä11] was developed as part of his PhD entitled "Eine modellbasierte Methode zur Objektivierung der Risikoanalyse nach ISO 26262" (free translation: A model-based method for objectifying risk analysis according to ISO 26262). The title of Ständer's PhD thesis directly addresses our overall challenge of providing a holistic model-based approach for HRAs. In the introduction, Ständer sets his scope of contributions to formalizing the terminology used in ISO 26262 and to deriving a model-based approach for situation definitions and assessments.
For the formalization, Ständer transformed the terminology of the HRA process into a petri-net (cf. Figure 2.8).

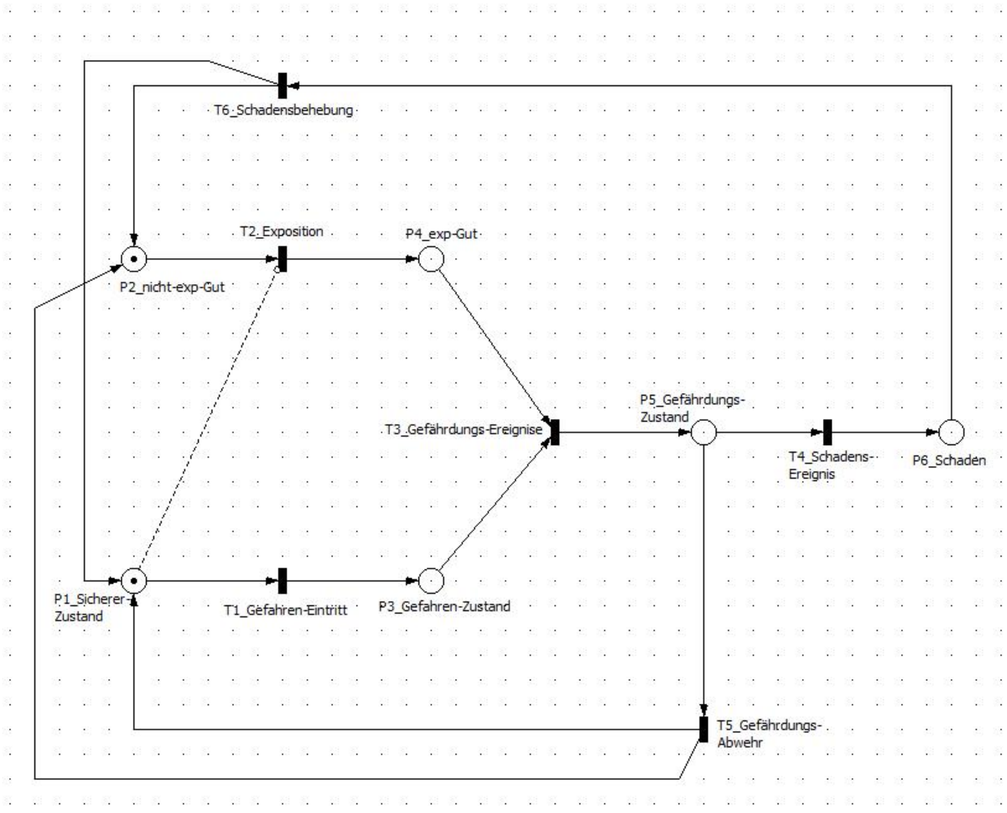


Figure 2.8: Ständer's HRA terminology translation into a petri-net

This approach is an adoption of the petri-net approach of Schneider from 2010 [WBS10]. The advantage of this approach is that it visualizes the relations and precisely defines contributing factors and the sequence of states and transitions to an accident. Using this formal model, he defines ISO 26262's parameters (Exposure, Controllability, and Severity) in terms of states and transitions in the petri-net model. By doing so, he can formally derive the contributing factors to consider when defining and assessing the ASIL parameters. With a reachability analysis he proves that the exposure parameter cannot be determined independently as ISO 26262 demands. This would be a highly relevant conclusion for our work. However, we do not adapt his view on exposure because he (re-)defined "exposure" as being the probability of the hazardous situation, "P5_Gefährdungs-Zustand" (cf. page 93 in chapter 6.1.1 of his PhD). ISO 26262, however, strictly distinguishes between hazardous situation and the "operational mode and environmental situation", where the latter gets the exposure parameter. A hazardous situation is the concomitance of hazard and situation.
His second contribution is the formalization of exposure. For this formalization, he drops his finding mentioned above and limits himself to the notion of exposure as given in ISO 26262. For the formalization, he uses a decomposition approach developed by Meyer in 2004 [MzH04].

As shown in Figure 2.9 Ständer distinguishes between the environment and the vehicle itself and provides generic refinement levels for these classes. However, there is no description of how to instantiate this meta-model and derive situations. We additionally interpret ISO 26262's "operational mode" as describing interaction or maneuver information. For us, it is not clear whether or not this concept is contained in the meta-model or in which meta-element this is subsumed.

Ständer uses the meta-model as a basis for his petri-net approach for determining the probability of situations. In order to do this, one has to model a petri-net of relevant influence factors and their possible relations. For each transition in the petri-net, one has to define a kind of conditional probability, where the condition is given by the preceding path in the petri-net. This approach has the advantage that the situations in a petri-net are specialization-consistent. The real specialization of a situation is an extension of a given petri-net path. This naturally results in an equal or lower probability (because each additional transition mathematically adds a factor between 0 and 1 to the probability). The disadvantage of this model is, however, that modeling a petri-net for each situation analysis is an enormous effort. From practical experience we would assume that this is too time-consuming for practical application. Another disadvantage is that the real difficulty lies in defining the transition probability in the model. This is especially true since we need precise transition values, but in the end the exposure parameter is classified by orders of magnitude. The preciseness of the model is therefore very time-consuming on the one hand and not necessary on the other hand. Besides this, it is still up to the engineer to decide which situations to model and how fine-grained the situations are. A key result presented in the thesis

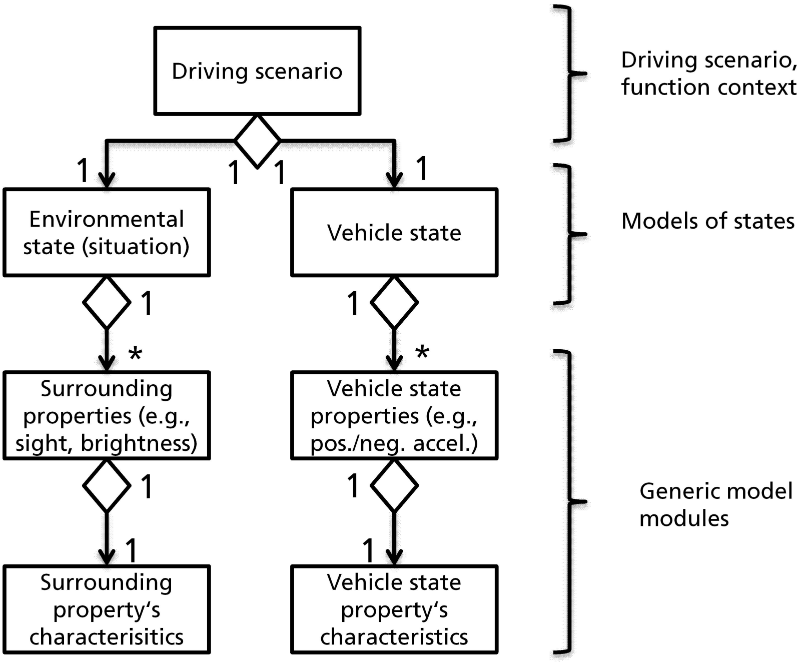

Driving scenario
1
1
1
1
Environmental state (situation)
Vehicle state
1
1
*
*
Surrounding properties (e.g., sight, brightness)
Vehicle state properties (e.g., pos./neg. accel.)
1
1
1
1
Surrounding property's characterisitics
Vehicle state property's characteristics
Driving scenario, function context
Models of states
Generic model modules

Figure 2.9: Meyer's influence factor decomposition approach (translated from German to English)

is that with the EmMORI approach, risk assessment tends to result in lower ASILs (due to the precise situation modeling). From our point of view, this is critical because no argumentation is provided that shows that the requirement from ISO 26262 ("situations shall only be as detailed as necessary") is fulfilled. Furthermore, there is no support for coming up with the petri-net transition probabilities and no consistency check is possible as to whether the final exposure value is correct or not.

Ständer's work focuses on overall formalization by using a petri-net formalization approach. Due to conclusions contradicting ISO 26262's view, his formalization approach does not provide useful information for our work. His second focus on situation modeling and exposure assessments based on petri-nets is interesting, but leads us to the conclusion that formalization of influence factors and the definition of influence factor probabilities is not a suitable approach and that formalizing situations in such a way that the probabilities can be calculated is not the most effective and efficient solution. Another huge area of related work is accident analysis. One of the most popular methods in this area is Leveson's STAMP (Systems-Theoretic Accident Modeling and Processes) approach [HWL06]. STAMP "rests on modeling and analyzing socio-technical systems and using the information gained in designing the socio-technical system." The goal is to analyze accidents and draw conclusions for the process of developing future systems. All approaches dealing with accidents post-factum naturally try to reconstruct situations and system failures. In contrast, we try to prevent accidents in the first place and focus on the prediction of possible situations and system failures. Especially the question about completeness cannot be answered with post-factum approaches, because the investigation focuses on one situation, i.e., the concrete situation that existed at the time of the accident. Accident investigations define "completeness" too, but their completeness focuses on the investigation of all contributing events to the accident (not all possible situations). Another fact is that we focus on functional safety, whereas accident reconstruction models usually deal with the whole process as a "socio-technical system"[HWL06]. We therefore omit this area of research from further investigation.

2.3.4 Conclusion

To the best knowledge of this author, there exists no further related work that directly addresses the improvement of the overall HRA process. Based on the reviewed work, we state that most of our core challenges are still open. The only challenge for which the state of the art if sufficient is the identification of system-level failures as describes in the ATESST2 discussion. Furthermore, an operational situation in the reviewed approaches always structured by classifying and structuring smaller building blocks of situation influence factors. How to come up with these influence factors, what a good structure is and how to use these building blocks to produce situations has not been answered.

2.4 Related Work

In this section, we will present related work that addresses partial challenges of our holistic model-based approach for structured HRAs. To do so, we stick to our scheme of challenges introduced above and discuss the related work corresponding to it. Please note that we omit the identification of system-level failures because we identified this as being solved in the previous section already. The remaining three main challenges are:

- Correctness of situation analysis
- Hazardous event classification
- Dealing with multiple service failures

2.4.1 Correctness of situation analysis

The first partial challenge we will discuss is the challenge of "***Formalization of the operational situations***". Besides the approaches introduced above, there is no other approach dealing with formalization of environmental information for application in HRAs in the automotive domain. There is, however, a lot of work that can be used to solve partial challenges. In the

following sections, we will discuss how it is related to our work and show which aspects can be (re-)used in our approach.
The challenge of modeling a system in its environment is not unique for safety engineering. Many areas deal with a system in its environment in some way. Since it is an early phase of development, requirements engineering is the most important area and has a similar understanding of the importance of context and environmental information. In 1997 Jackson stated [ZJ97]:

It is not necessary or desirable to describe (however abstractly) the machine to be built. Rather, the environment is described in two ways: as it would be without or in spite of the machine, and as we hope it will become because of the machine.

For safety, we could extend the last sentence by explicitly stating "as we hope it will *and won't* become because of the machine". Jackson's view on the environment is obviously broader than ours, but he directs the focus away from the system to the environment. The meaning for safety is expressed very nicely by Lutz [Lut01]:

It is not the internal complexity of a module but the complexity of the module's connection to its environment that yields the persistent, safety-related errors seen.

In 2008, as part of his PhD, Allmann investigated the state of the art for situation- and scenario-based requirements management in the automotive domain [All08]. He extracted the three main specification techniques for scenario-based requirements elicitation.

- Statecharts [Har87][DH89]
- Software Cost Reduction (SCR) [Hen80][vSPM93]
- Formal Specification Language (ForSeL) [HRWS06]

Allmann concluded that statecharts and the SCR method are used in practice and science but they are not suited well for abstract descriptions of scenarios and situations. Based on this finding, he used the ForSeL language because of its "intuitiveness and high usability"[All08]. Furthermore, he stated that by using ForSeL one does not exclude a higher level of formality and complexity because ForSeL can be translated and refined through more formal languages such as SCR. Allmann provides a meta-model for scenario modeling and extended the ForSeL approach to describe concrete scenarios. He did his PhD at AUDI AG and therefore strongly focuses on the needs of the automotive domain and on practical applicability. He explicitly mentions ISO 26262 in general and especially discusses HRA. He cites Grady [Gra93] by stating that safety engineering and requirements engineering are coupled too loosely. Allmann's work therefore integrates safety engineering in order to make it possible to represent safety scenarios in his framework as well. Furthermore, he defined the completeness of scenarios by using Boolean logic on relevant classes of influence factors. The logical constraints are defined by manually specifying rules stating which scenarios should be considered. With a predefined set of influence factors and the set of rules he can decide whether all subsets (all situations) have been considered. While this is generally a good indicator for requirements scenario completeness, it uses assumptions that are not applicable for our challenge. First of all, he assumes completeness of the set of relevant influence factors. For the requirements scenarios, this is applicable because the situations are more abstract and general than the ones for HRA. Thus, the set can be described efficiently. As discussed above, this is not possible for HRA influence factors. To find relevant influence factors, i.e., to know what is relevant and what is not relevant is however exactly the challenge we see as fundamental for HRA. The other assumption is the completeness of the rule set. Due to the definition of being complete, iff all rule-derivable scenarios are considered, implies that the rule set has to be complete (and correct). Defining rules capturing the complete domain knowledge for the situation definition of HRAs is not possible (as will be discussed as part of the situation reuse discussion).

A fundamental observation is that the environment cannot (efficiently) be described without the system and vice versa. Thus, the most basic concept we need is an integrated model of the system in its environment. One of the most influential models addressing this challenge is Parnas' and Madey's Four-Variable Requirements Model from 1995 [PM95]. The idea of Parnas and Madey was to provide a more formal way to describe the requirements of complex software systems. It is named Four-Variable Model because they defined four sets of variables and relations between these variables.

Monitored: These are hardware inputs from the physical world.

Controlled: These are hardware outputs into the physical world.

Input: Variables in this set are environmental quantities expressed in the logical world.

Output: Variables in this set are environmental quantities expressed in the logical world.

The defined relations are:

IN: This maps input variables to monitored variables; IN describes the transition from physical values to logical values.

OUT: This maps controlled variables to output variables; OUT describes the transition from logical values to physical values.

NAT: This relation places constraints on the values of environmental quantities. NAT describes the environmental context of the system that is being controlled.

REQ: This relation places further constraints on the environmental quantities. REQ describes the behavior of the system in its environment.

SOFT: This relation describes requirements for the system development. SOFT is of no further interest for our challenge.

Note that the approach was meant for dealing with software-systems and not for systems engineering. Miller and Tribble, however, "extended the Four-Variable Model to bridge the System-Software gap" [MT01]. Galloway et al. extended the FVM in order to formally develop safety-critical software [GIMT08]. The basic principle of separating the physical from the logical domain is given in both extensions and we therefore omit the explicit explanation of this model here. Especially noteworthy is the concept of separating the logical from the physical world. An accident and direct sources of harm are always located in the physical world. Functional safety, however, addresses unwanted behavior in the logical world. Therefore, the Four-Variable Model forms an important conceptual basis for our work by bridging the gap between the logical and the physical world.

Another more formal starting point for environmental formalization is the area of mathematical situation theory.
"It is an information theoretic mathematical ontology developed to support situations" [Dev95].

Rather than trying to represent total possible worlds, situation semantics is a relational semantics of partial worlds called situations. As such, situations support (or fail to support) items of information or in terms of situation theory, infons. They are of the form

$$\langle\langle R, a_1, \dots, a_n, 1\rangle\rangle, \langle\langle R, a_1, \dots, a_n, 0\rangle\rangle \tag{2.6}$$

where R is an n-place relation and a_i are objects appropriate for R. The last parameter indicates the polarity, whether the relation holds or does not hold. In our context, examples of infons are:

$$\sigma_1 = \langle\langle \textit{excludes, Highway, parking, 1}\rangle\rangle \tag{2.7}$$

$$\sigma_2 = \langle\langle \textit{excludes, Highway, parking, 0}\rangle\rangle \tag{2.8}$$

Thus, we assume we have a relation called "excludes". The semantics are that it relates certain aspects of driving situations to each other. In this example we have created two infons. One states the fact that "parking on highway" is not possible (= excludes with polarity 1) and the other one states that it is possible (polarity is 0). One would assume that either parking on a highway is possible or not. Stating both facts seems to be kind of inconsistent. This is of course true, but it is true for one specific "world". Due to the fact that the world is described by situations, we can state that we have to define which infons make sense in a situation and which do not make sense. When a situation supports a state of affairs, we say that the situation "s" makes the concrete infon "σ" factual, written $s \models \sigma$.

Due to the very high and theoretical level of abstraction, we do not want to discuss situation theory in detail. A brief and non-technical introduction to situation theory and situation semantics can be found in Devlin's work from 2006 [Dev06]. Due to the formality of situation theory, it makes many aspects explicit, which is very interesting for our approach. Especially the separation of infons and situations is fundamental. Many approaches, such as the one we discussed above, use partial aspects or facts in order to construct a situation. These facts such as "city driving" are all similar to the concept of infons in situation theory. The difference, however, is that situation theory has a formal background and is therefore more conclusive. It does therefore make sense to align our approach with the general concepts of situation theory. Due to its rather elementary nature, however, the transition from basic mathematical research results to concrete application in environment formalization for HRA is not trivial.

Situation theory constitutes a good formal basis for describing all kinds of situations. Due to its abstract nature it does not help in answering questions about how to construct a good domain-specific situation theory, i.e., it does not help to identify infons and does not support the decision regarding the identification of necessary and sufficient criteria for creating situations. For HRA, however, one important part of the "Formalization of operational situations" is to know what information is necessary and sufficient to describe a situation suitable for an HRA process. Thus, we need to move one step further to practical applications of situation theory or of environment descriptions. In this direction of research the concept of "context" appears. With their work "The use of situation theory in context modeling" [AS97] Akman and Surav support a formal transition of situation theory to context modeling. Their work focuses on context in natural language and logical AI. They cite a context definition of Crystal from 1991: "Context is a general term [...] to refer to specific parts of an utterance near or adjacent to a unit which is the focus of attention" [Cry08]. With this notion, they formalize the relation of *context* and *situations* mathematically. The fundamental difference between context and situation is the "presence of a unit". Thus, context always refers to some application or unit, while a situation does not have this constraint. This notion is congruent with the natural language usage of "context" and "situation". When we talk about context only, we usually ask "context of what", while a situation can stand on its own; it is self-contained in the sense that it contains all information to make a useful statement. With this distinction we could state that a situation is a wider concept than context. Describing the context of something could be defined as eliciting all situations relevant for the unit the context focuses on.
For our work, the general term situation is obviously limited to the aspect of being relevant in the usage of a vehicle. Thus the situations we are looking for or the environmental aspects we want to formalize are in the context of vehicle usage.
Looking at the domain of context modeling, a plethora of related work appears. For our work, most of it is not essential because most of the work focuses on the concrete modeling of a context for a specific application scenario. Due to the fact that we have not found related work addressing context modeling in support of the automotive HRA process, we will not discuss concrete context modeling approaches, but will rather discuss context modeling on a meta-level in order to derive essential concepts for our approach. In order to understand "context" better, we would like to refer to

the work of Abowd et al. [ADB$^+$99]: "Towards a Better Understanding of Context and Context-Awareness" from 1999. In this work, they give a more detailed definition of context:

"Context is any information that can be used to characterize the situation of an entity. [. . .]"

The concept of characteristic information of an entity is obviously essential in context modeling. For our approach, essential information means how to represent "context". Strang and Linnhoff-Popien published "A Context Modeling Survey" in 2004 [SLP04]. This survey focused on the applicability of context modeling approaches for ubiquitous computing. They introduce six quality attributes of context modeling approaches.

1. *Distributed composition (dc)*: As ubiquitous computing systems are usually distributed computing systems it is important that the context models are composable. For our approach, the distributed composition attribute is of minor importance.
2. *Partial validation (pv)*: This quality attribute describes the possibility of validating contextual knowledge on the structure as well as on the instance level. For HRA, this is obviously highly important because it enables consistency checking of captured knowledge.
3. *Richness and quality of information (qua)*: The quality of information sensed in the environment varies over time. For ubiquitous computing, the tagging of context information wrt. quality is important. For HRA, however, we have ideal situations and are not dependent on any sensor quality. Thus, this quality attribute is irrelevant for us.
4. *Incompleteness and ambiguity (inc)*: This attribute addresses the fact that the knowledge model may be incomplete and ambiguous. A model of context knowledge should be able to handle this. Incompleteness, in particular, is highly important in our approach because we cannot include every single peculiarity of an environment.
5. *Level of formality (for)*: Formality is obviously important in most applications. It enables the precise and automated handling of data on the one hand and resolves ambiguities on the other hand. Due to our goal for HRA, a certain formality is highly important.
6. *Applicability to existing environments (app)*: This attribute addresses "legacy" ubiquitous computing environments in the sense that every new context modeling approach should support the existing environments. This is obviously not of interest for us.

Because ubiquitous computing is quite distinct from situation modeling for automotive HRA, we cannot use their evaluation of applicability, but we have to come up with our own rating based on the essential quality attributes for context wrt. automotive HRAs. The quality attributes "distributed composition", "quality of information", and "applicability to existing environments" are of minor interest to us. The reason is that we are in an idealized situation

and solely in the design time. If we had to perform a kind of runtime HRA, distributed composition and quality of information would most certainly be indispensable.
With these quality attributes, Strang and Linnhoff-Popien evaluated six context modeling approaches:

a) Key-Value Models
b) Markup Scheme Models
c) Graphical Models
d) Object-oriented Models
e) Logic-based Models
f) Ontology-based Models

The result of their survey is compactly given as an appropriateness indication table (cf. 2.8).

Table 2.8: A Context Modeling Survey Table [SLP04]

Approach-Requirem.	*dc*	*pv*	*qua*	*inc*	*for*	*app*
Key-Value Models	-	-	-	-	-	+
Markup Scheme Mod.	+	++	-	-	+	++
Graphical Models	-	-	+	-	+	+
Object Oriented Mod.	++	+	+	+	+	+
Logic Based Models	++	-	-	-	++	-
Ontology Based Mod.	++	++	+	+	++	+

Note that we grayed out the columns not relevant for us. The result is that the most promising approach for describing context or, in our case, environmental models is an ontology-based approach. In the solution chapter, we will further refine how we used ontologies to support our approach.

Concluding the related work review on "Formalization of operational situations", we state that:

– The Four-Variable Model provides a very good conceptual basis for our work by bridging the gap between the logical and the physical world.
– Situation theory gives a formal background for dealing with situations and relevant pieces of information, called infons. Our approach will be based on the idea of situation theory.
– Context modeling is a promising area from which to adapt concrete modeling approaches. The survey for context modeling has shown that the most suitable modeling approach for contexts or environments (for HRA) are ontology-based models.

We still have to answer to the question of what constitutes necessary and sufficient information for an HRA. Besides the general guideline from the ISO 26262 standard saying that one has to make sure that a situation contains only necessary but sufficient information, we could not find any related work addressing the question of what exactly this means.

The next area of related work addresses the challenge of "***Efficient reuse of operational situations***". Formalization is a prerequisite for efficient reuse. As we have seen in the previous section, there is no approach available for the formalization of situations for HRA. Thus we could not find any efficient reuse approach in the related work. If we abstract from the demand of being explicitly suitable for situations in the HRA context, we find ourselves in the area of "efficient reuse of knowledge". Ruben Prieto-Diaz discussed in his keynote talk at the 2001 Symposium on Software Reusability the general nature of reuse by comparing the traditional engineering and the software engineering disciplines [PD01] . Most interestingly, he stated that there is no reuse sub-community in traditional engineering. Electrical engineers, for example, do not have special courses on reuse. For software he argues that "despite several years of trying to bring reuse to practice, software engineers have found out that reuse in software is not the same as in other areas, that software is very hard to reuse." As he points out, the fundamental reason is that software engineers always try to find the best possible solutions. Traditional engineering disciplines usually try to find the best trade-off. Thus, traditional reuse does not work for software because we expect every component to be perfectly reusable without any adaptation. For our goal, it is important to have precise situations on the one hand, but on the other hand we do not only want to reuse identical situations, but we want to reuse similar situations as well (as traditional engineering would do).
Another requirement for efficient reuse in our context is practical applicability. In general, there are two extreme approaches for reuse in practice. One is to build a model covering all possible reuse situations up front. This is a kind of investment for reuse. Its advantage is that we can expect a very high reuse factor, but the drawback is the initial high investment in building the repository. The other strategy is to incrementally build up our reuse repository by storing more and more concrete artifacts. In this strategy, the reuse possibilities increase over time. The advantage is, of course, that we do not need any significant up-front investment; the disadvantage is the initial time where we have to put information into the repository without any significant reuse potential.

In our case, as discussed earlier, it is not efficiently possible to come up with a set covering all possible situations. Therefore, the first strategy is not suitable for us. The second strategy seems to be more appropriate, because we already have some situations from the ISO 26262 standard and from industrial projects. Thus, it is a good idea to use the second strategy but skip the initial learning phase by filling the reuse repository with an initial set of situations.

The important aspect in reusing situations, however, is not to avoid creating the same situations over and over again, but to reuse the related domain knowledge. The essential domain knowledge for HRA is not only the situation itself, but particularly the exposure probability assigned to the situation. To recapitulate the idea of exposure: The exposure parameter is the assessment parameter defining an order of magnitude of the likelihood of being in a certain driving situation. The scale ranges from E0 to E4.

As discussed in the previous section about the formalization of the environment, situations usually consist of partial aspects. These aspects have different names, ranging from infons to influence factors. In the following, we will use influence factors, but independent of the term, the notion is always to have atomic information artifacts. A situation is therefore a set of relevant influence factors together with an assessment of the probability of the situation. It is appealing to base the reuse approach not on situations, but on the more basic building blocks, the influence factors. Due to the fact that a situation is a set of influence factors we have an exponential number of situations we can derive from a set of influence factors. Thus, if we could assign atomic independent influence probabilities, we could make the reuse process even more efficient. However, this is not possible because the probability of influence factors is highly dependent. Imagine the influence factor "ice": The probability of ice occurring on the road is obviously higher if the influence factor "-5°C" is given (than in a situation where we have "+20°C"). Still, we could try to achieve this, but we have to deal with conditional probabilities.

The technological solutions in this direction are probabilistic graph models with machine learning or rule inference and induction approaches.

Both approaches try to infer new knowledge based on given facts. For conditional probabilities, Bayesian networks in particular, seem to be promising [Bis07]. For a Bayesian networks approach we see the following challenges: If we want to infer information based on the building blocks of situations (the influence factors), we would need a lot of information about the influence factors and their dependencies with respect to conditional probabilities. This would either require explicit modeling of dependencies between influence factors or we would need to learn from examples. Thus, we would have to "learn" from the situations and their probabilities and derive conditional probabilities for influence factors. This is very difficult to achieve because we have only a very limited number of situations and their probabilities to learn from. For good "training", we would need a very high number of training situations for the approach to learn.
Another disadvantage is the greedy learning action. This means that the techniques do not wait for a new problem or target to be solved, but they try to infer as much as they can up-front and if a new target comes up, they simply try to match their "knowledge" to the target. If the number of situations increases, the "knowledge" has to be updated completely every time. For us this would mean that we would have to derive all possibly constructible situations, which would only create a huge amount of data, which might not be necessary. This is obviously not efficient. Another challenge is that we have to adapt traditional approaches in order to deal with two special conditions:

1. We need to deal with an open world assumption on situations. The open world assumption essentially implies an intuitionistic logic, which means that we cannot derive "false" if if something is "not true". So the basic law of many logic systems, "tertium non datur", does not hold. For example, not stating "ice" in a situation does not mean that there is definitely no ice.
2. The other special condition is that we want to have our "world" extendable in the sense that we want to be able to add details about influence factors and add influence factors, meaning the "world" might change occasionally.

Due to the arising challenges and the disadvantages, our evaluation of using probabilistic graph models for influence factor based models is that these solution areas are not appropriate for our application scenario. For us it seems to be more appropriate to use the situations directly as reuse assets. We want an approach without the overhead of eagerly trying to infer as much knowledge as possible.

Case-based reasoning is highly appropriate for doing this. Its basic idea is to use old cases and try to derive knowledge (lazily) if a new case comes up [Lea94]. For us the cases are situations and if a new situation comes up we want to compare this situation to older situations. Hence, this fits well to our problem. Furthermore, case-based reasoning does not need any up-front investment in cases, meaning there is no minimal required set of cases to allow it to be used. Note that having some cases upfront is good, of course. For our approach, this is the optimal prerequisite.

The process of case-based reasoning is described in [AP94] as a four-step approach:

1. RETRIEVE cases that are similar to the new problem case
2. REUSE the information of the retrieved cases to solve the problem
3. REVISE the new solution in order to check whether it is valid or not
4. RETAIN the new case in the general knowledge database to support future problem solving tasks.

This process is depicted in Figure 2.10.

This technology fits very well into our situation-reuse challenge. The "problem" in our case is to determine the exposure of a new situation. The new situation is therefore our "new case". By retrieving similar situations that have already been assessed, we can suggest an exposure value for the new situation or at least give constraints for consistently assessing the new situation. The developer or domain expert then revises the suggestions or the constraints and assigns the final exposure to the new situation as a confirmed solution. Having the expert in the loop has the additional advantage that all decisions are made by the expert. Especially in the area of functional

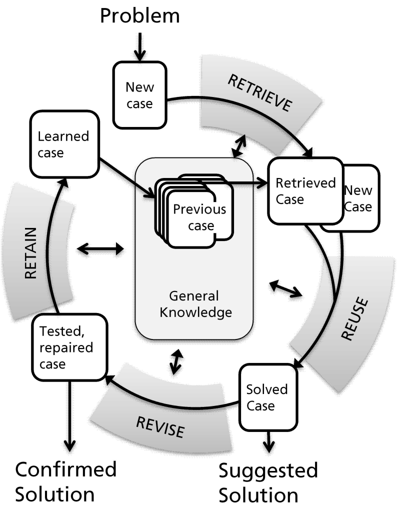


Figure 2.10: Case-based reasoning cycle [AP94]

safety, this has great advantages due to responsibility reasons. The "case" is afterwards retained in the knowledge base.

Thus, case-based reasoning is a very good approach for our "Efficient Situation Reuse" challenge, but we need to define what similar situations are and how to represent situations as cases. As discussed in the previous section, our influence factors and situations are modeled in an ontology. Recio-Garcia presented in [RgDaGc$^+$06] the connection between ontologies and case-based reasoning. The basic idea is to have the ontology as the general knowledge database and define similarity metrics on the structure of the ontology. This is generally a very nice approach, but we still need to adapt this idea to the application scenario dealing with situations and their associated influence factor ontology.

Overall, we can conclude for the section on "Efficient Situation Reuse" that there exists no specific related work for HRA-focused reuse of situations. In the general area of reusing knowledge and infering new knowledge from already existing knowledge a huge amount of research work is available. The focus of this thesis is not on advancing the state of the art of reuse and knowledge engineering in general. Therefore, we evaluated the current state of the art with the focus on applicability for our challenge. The most appropriate and best fitting related work in this area is the use of case-based reasoning in combination with ontology engineering.

With this observation, we also finish our discussion of related work with respect to the challenges comprising the completeness of situation analysis.

2.4.2 Hazardous event classification

Hazard classification is essentially the assessment of three risk parameters:

- Exposure
- Controllability
- Severity

The assessment of the ***exposure*** parameter deals with the estimation of the likelihood of a driving situation. While there exists related work for the description and definition of situations as seen above, the classification of the likelihood is specific for the automotive domain (and for the ISO 26262 standard). Besides the above-mentioned approaches discussed as part of the state-of-the-art chapter, there exists to the best of our knowledge no related work that specifically addresses exposure assessment.

The parameter "***controllability***" puts the human into the chain of actions when considering an accident or an incident. ISO 26262 defines controllability as the

"ability to avoid a specified harm or damage through the timely reactions of the persons involved, possibly with support from external measures".

In this definition, we can identify two main aspects that have to be considered:

1. the ability to avoid [...] harm [...] possibly with support from external measures, and
2. timely reactions of the persons involved

The first aspect refers to changing the use of the subsystem that has a failure. The appended "external measure" especially points out that controllability is often based on using other functions to compensate for a failure. If, for example, the brake system has a failure and an unwanted lateral acceleration occurs, this may be controlled by the driver by increasing or decreasing the steering angle. For a controllability argumentation using the integrity and availability of another function one obviously needs to show this. The underlying challenge refers back to the challenge discussed above of dealing with multifunctional degradation. While this will provide the technical dependency information, controllability will reuse this knowledge to prove the first aspect of a controllability argumentation.

The second aspect of controllability is the description of the reaction of a person involved in a particular scenario. Not only the correct reaction, but also the timely execution of the reaction needs to be argued. As already discussed in the problem statement, the formalization of this aspect would lead us into the area of human factors and human capability modeling and will not be considered in this thesis. We discussed some fundamental issues of human factors in the context of ISO 26262 as part of our publication [KT11].

The interested reader is referred to this publication for further references that might be interesting in this area.

The assessment of the ***severity*** of an accident is the *"estimate of the extent of harm to one or more individuals that can occur in a potentially hazardous situation"* [ISO11].

In ISO 26262, the Abbreviated Injury Scale (AIS) is used to classify the severity of an injury. This scale has seven classes: AIS 0: no injuries to AIS 6: extremely critical or fatal injury. In Annex B of part 3 of ISO 26262, it is stated, however, that other categorizations could be used. The rationale is that ISO 26262 acknowledges that the use of AIS can only classify single injuries. Multiple injuries of one person or injuries of multiple persons cannot be classified with AIS. The aggregation of injuries is, as stated in the problem statement, still a field of research. As alternatives for AIS, ISO 26262 refers to MAIS and ISS, which we will present very briefly.

MAIS stands for maximum abbreviated injury scale [HJE+10]. The straightforward approach is to select the maximum of the AIS scores. Note that a "maximum" operation requires at least an ordinal scale. The seven-class version of the AIS as used in ISO 26262 meets this requirement. The original AIS code, however, is a nominal scale because there is an additional code 9 assigned to "not further specified injury". The MAIS approach is probably the implicit approach one applies in case of multiple injuries. The second approach mentioned is the Injury Severity Score (ISS) [BOHL74]. This approach originated in polytrauma research, thus considers mostly severe injuries. The basic idea is to convert the ordinal scale AIS codes into survival probabilities. These probabilities are then interval-scaled and enable arithmetic operations on the values. This is a neat handling of the problem of not being able to calculate an overall injury score from single injuries. Some objections should be pointed out as well. The first one is the independence of the injuries. This independence is required in the arithmetic's of ISS calculations. The second objection is the way the calculation is defined. It is based on squaring and adding single injury probabilities. This is based on empirical data collected for polytraumatic estimations, not necessarily suitable for usage scenarios in the automotive domain. Therefore, there is still ongoing research in this area. The precise and formal decision of the severity of an accident is not within the scope of this thesis, however, and we will therefore omit further discussions of related work in this area.

2.4.3 Dealing with multiple service failures

The last challenge we have to investigate is how to deal with ***multiple service failures***.

The assessment of a multiple service failure itself is not different from the assessment task of a single service failure. The challenge for the assessment arises from the number of multiple service failures. Manually assessing an exponential number of hazards (2^n) is obviously not possible. Automation or computer-aided analyzes do not exist (to the best knowledge of the authors). However, one could claim that in avionics, multiple service failures are handled with their functional hazard analysis (FHA). This true, but the advantage of avionics systems (in the past) has been that the systems comprise many single independent sub-systems. Thus, strong interrelations and dependencies did not occur by construction. In the future, avionics systems will rely on highly integrated architectures, implying the need for a more structured and formal way of dealing with multiple service failures. This trend is already becoming evident when we look at avionics industry demands in current research projects, such as SPES XT. An explicit task for developing a model-based, structured approach for FHA is strongly demanded by the avionics industry. Thus, we conclude that the current FHA is not applicable for our approach.

We can summarize the review of related work for ***dealing with multiple service failures*** by stating that with failure logic models, a good basis for addressing the challenges of multiple service failures already exists. Furthermore, there are already existing analysis techniques (such as FHA) for dealing with multiple service failures. There exists, however, no approach that focuses on relevant multiple service failures and deals with efficient assessment.

3 SAHARA - a Structured Approach for Hazard Analysis and Risk Assessments

In the previous chapters, we gave an introduction to hazard analysis and risk assessments and derived a structure of challenges and goals. Subsequently, we gave an overview of the current state of the art and work related to structured model-based hazard analysis and risk assessments.

In this chapter, we present our contributions and solutions for the identified challenges and goals. The structure of this chapter is such that we will first give a solution overview, which explains the overall solution architecture, the building blocks, and their role in SAHARA. In the subsequent sections we present the concrete solutions and contributions of the four main building blocks of SAHARA, namely GOBI, OASIS, HEAT, and ARID.

3.1 Solution Overview

In this section, we will explain SAHARA's solution architecture. Figure 3.1 depicts an overview of our solution.

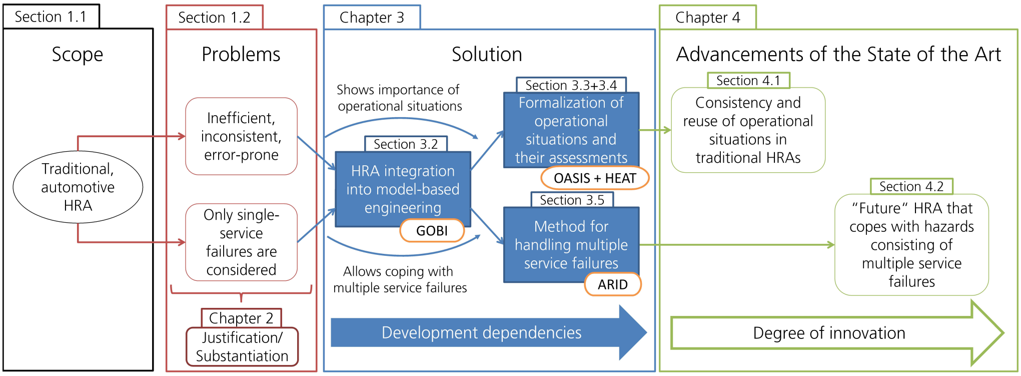

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.2
HRA integration into model-based engineering
GOBI
Section 3.5
Method for handling multiple service failures
ARID
Allows coping with multiple service failures
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 3.1: PhD overview of SAHARA's solution

Please note that we combined the OASIS and HEAT blocks into one solution block because both address the formalization of operational situations. Corresponding to the HRA process, which distinguishes between hazard analysis and risk assessment, we separated the aspect of defining operational situations from the aspect of assessing operational situations. Therefore, we refine the one block combining OASIS and HEAT into these two aspects as shown in Figure 3.2.

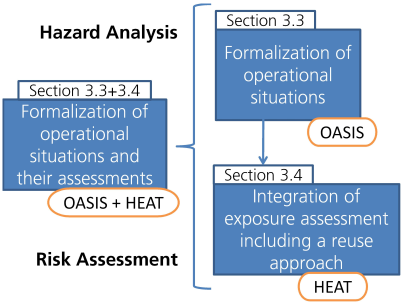

Hazard Analysis
Risk Assessment
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.3
Formalization of operational situations
OASIS
Section 3.4
Integration of exposure assessment including a reuse approach
HEAT

Figure 3.2: Refinement of the operational situation formalization

In this section, we translate our solution overview, shown in Figure 3.1, from a problem-driven perspective to a solution-oriented perspective. Figure 3.3 shows on the left the problem-driven coarse-grained structure and on the right our detailed solution architecture. The latter structures the solution building blocks as discussed in the next sections and shows their dependencies.

In the introduction, we already discussed that the model-based representation developed in GOBI is a common foundation for our core contributions presented in the sections on OASIS, HEAT, and ARID. Based on our problem discussion and the formalization in GOBI, we derive specific technical challenges for the respective contribution blocks to solve. These technical challenges are represented as orange ovals. Hence, in order to reach the goal shown on the left of Figure 3.1, we had to cope with the challenges shown as corresponding orange ovals on the right. For example, the block on the very left *"Formalization of operational situations"* is the goal of OASIS. In order to reach this goal, we basically have to solve the two technical challenges, *"Formalize and structure situation restrictions"*, and *"Correctness of situations (wrt. concomitance)"*. The detailed blocks on the lower part of the figure that shows the solution domain, depict the sections we are going to discuss in the subsequent solution approach.

However, before discussing the detailed solution blocks, we will explain the solution architecture on a black-box level. Due to the black-box view, we will focus on the building blocks as such and especially describe the connections shown in solution part of Figure 3.1.

We will use an interface specification with the following recurring structure:

- State the main challenge of the building block
- Explain the provided interface.
- Explain the required interface.
- State the result as a concrete *contribution*.

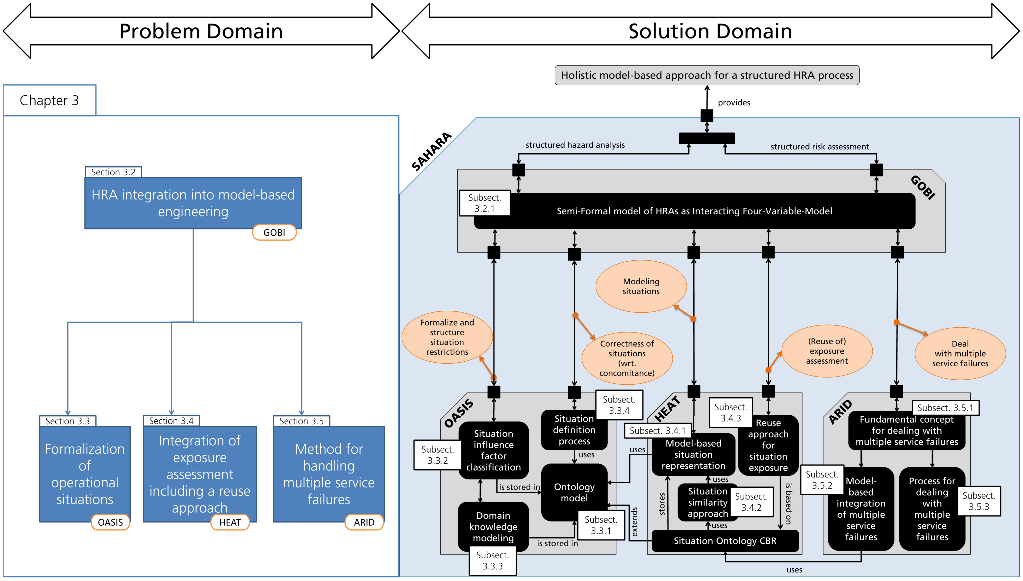

Problem Domain
Solution Domain
Chapter 3
Section 3.2
HRA integration into model-based engineering
GOBI
Section 3.3
Formalization of operational situations
OASIS
Section 3.4
Integration of exposure assessment including a reuse approach
HEAT
Section 3.5
Method for handling multiple service failures
ARID
Holistic model-based approach for a structured HRA process
provides
SAHARA
structured hazard analysis
structured risk assessment
GOBI
Subsect. 3.2.1
Semi-Formal model of HRAs as Interacting Four-Variable-Model
Modeling situations
Formalize and structure situation restrictions
Correctness of situations (wrt. concomitance)
(Reuse of) exposure assessment
Deal with multiple service failures
OASIS
Subsect. 3.3.4
Situation definition process
Subsect. 3.3.2
Situation influence factor classification
uses
is stored in
Ontology model
Subsect. 3.3.1
Domain knowledge modeling
is stored in
Subsect. 3.3.3
HEAT
Subsect. 3.4.1
uses
Model-based situation representation
Subsect. 3.4.3
Reuse approach for situation exposure
uses
Situation similarity approach
Subsect. 3.4.2
uses
stores
is based on
extends
Situation Ontology CBR
ARID
Subsect. 3.5.1
Fundamental concept for dealing with multiple service failures
Subsect. 3.5.2
Model-based integration of multiple service failures
Process for dealing with multiple service failures
Subsect. 3.5.3
uses

Figure 3.3: Overview of SAHARA in the problem domain and the solution domain

↓ GOBI

Challenge: The overall challenge and main contribution of SAHARA is to provide a holistic, model-based approach for structured HRAs.

Prov.-IF: GOBI subdivides the HRA process and our challenges into the two fundamental challenges: structured risk assessment, and a structured hazard analysis. This is the provided interface as shown on top of the GOBI box in Figure 3.3. The refinement of these challenges is done in GOBI using a conceptual model and propagated down to the single building blocks as concrete challenges.

Req.-IF: GOBI defines and formalizes the general participants of an HRA: the system, the human, and the environment. Using this conceptual model, GOBI revisits the HRA process and defines responsibilities for the three participants on the one hand and derives from that overall five concrete challenges for the subsequent building blocks as required interface.

GOBI uses Parnas' commonly known conceptual model, the Four-Variable-Model and extends this into an Interacting-Four-Variable-Model. The interacting participants are the core elements of an HRA: the system, the human, and the environment. The fundamental idea of (re-)using Parnas' model is the separation of the physical domain from the logical domain. We discuss in GOBI that this fits very well to HRAs. Using the GOBI-FVM approach the HRA is revisited and the two basic tasks, the hazard analysis and the risk assessment is explained using the FVM. By this we transformed the high-level process-based HRA to a model-based HRA requirements model. The requirements are demands on the the interaction as a whole and the respective single FVM in particular. These conceptual requirements are then propagated down to the single building blocks representing our core contributions, OASIS, HEAT, and ARID.

Contrib. 1 ***Formalized HRA framework:***

Semi-formal framework that bridges the worlds of HRA and systems engineering to allow talking more structured about HRAs.

↓ OASIS

Challenge: GOBI gives an overview of the basic requirements of a formalized HRA. One requirement of GOBI and the challenge addressed by OASIS is the formalization of operational situations.

Prov.-IF: OASIS provides an ontology-based structuring and classification of possible influence factors, i.e., situation restrictions that enable the formal construction of situations. OASIS furthermore provides a situation definition process that fulfills GOBI's requirements for situation correctness.

Internally, OASIS provides an ontology containing influence factors and corresponding domain knowledge.

Req.-IF: OASIS deals with influence factors and their construction into correct situations. The definition and formalization of "correctness" and the information on what an HRA relevant situation is is given by GOBI.

Contrib. 2 ***Ontology-based situation influence factor classification:***

A situation ontology for classifying influence factors and integrating HRA-relevant domain knowledge in order to enable a more formal and structured way of constructing situations.

↓ HEAT

Challenge: While OASIS structures the influence factors of an operational situation and describes a process for constructing a situation, it does not address the modeling and consistent assessment of operational situations wrt. their exposure. The corresponding challenge addressed in HEAT is how a situation along with its exposure can be modeled as reusable domain knowledge.

Prov.-IF: HEAT provides a model-based representation of operational situations including their assessment result, the exposure, and the rationales for that result. HEAT furthermore provides an analytic consistency checking approach as well as constructive consistency assuring assessment support. With this, HEAT provides a concrete solution for the requirements elicited in GOBI.

HE

Req.-IF: HEAT requires the ontological model of influence factors provided by OASIS.

Contrib. 3 ***Situation modeling and exposure assessment support:***

Model-based situation representation with a similarity function, for reusing the assessment results of (similar) situations, in order to support the safety engineer in consistent situation assessments.

OASIS formalizes situation influences and provides a situation definition process, and HEAT complements this with a modeling and reuse approach for operational situations. The other aspect making up the essence of an HRA, the hazardous event, is the hazard itself.

↓ ARID

Challenge: The corresponding challenge is how to deal with hazards consisting of multiple service failures and the resulting exponential number of hazards.

Prov.-IF: ARID provides a model-based representation of hazards consisting of multiple service failures along with their foothold within a system and failure propagation model. Based on that, ARID provides a process for dealing with multiple service failures through the extensive use of system and domain knowledge collected.

Req.-IF: ARID uses the situation assessment and support facilities of HEAT on the one hand and fulfills the demands of GOBI to not distinguish between single and multiple service failures when it comes to risk assessments of hazards.

Contrib. 4 ***Approach for dealing with multiple service failures:***

A model-based representation of multiple service failures and a corresponding method defining a computer-aided analysis process.

With the description above we gave a very condensed overview over our solution approach and our contributions. A more detailed overview and the internal goals of the respective solution components are presented at the beginning of each component section.

3.2 GOBI - Gradation of Baneful Influence

As stated above, GOBI constitutes a formalization approach of HRAs on the one hand and as such constitutes an integration framework for the core challenges identified above on the other hand. Figure 3.4 shows the integration of GOBI into the overall thesis.

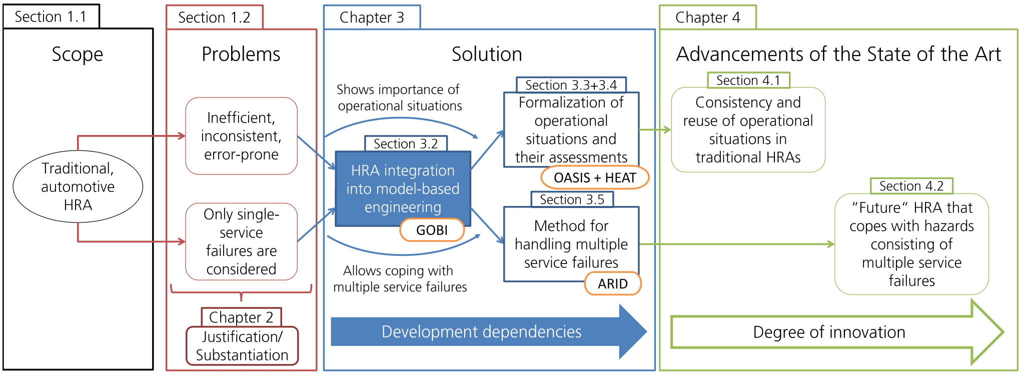

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method for handling multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 3.4: Integration overview of GOBI

For our Structured Approach for Hazard Analysis and Risk Assessments (SAHARA), we identified in section "2.2 Core Challenges and Goals" the following fundamental challenges:

✧ Holistic model-based approach for structured HRA processes

➠ Structuring of hazard analysis

➠ Structuring of risk assessments

Additionally, we derived from our state-of-the-art review that there is no integrated view or conceptual model of HRAs. With GOBI we provide such a holistic, model-based integration framework, by mapping the general idea of an HRA to a conceptual, but integrated model. This allows us to provide a semi-formal integration framework for the challenges elicited in chapter 2.2. The core challenges that are integrated into this framework will be the topics of subsequent sections called OASIS, HEAT, and ARID.

We structure this section in the following way:

1. **The Gradation - Understanding the overall idea of an HRA**
   In this section, we will first discuss the HRA and derive our conceptual, semi-formal system representation as Interacting Four-Variable-Model (FVM). The purpose of this model is to bridge the world of HRAs and systems engineering.
2. **The Systems – Detailing the core elements**
   In this section, we will investigate the interacting FVMs in more details.

We will discuss properties and peculiarities which need to be considered in the subsequent sections.

3.2.1 The Gradation – Understanding the overall idea of an HRA

As the title implies, the goal of this section is to provide an overall framework for formalized discussions of HRAs. The challenge of such a formalization is to connect the two worlds of HRA-focused safety engineering and systems engineering. We approach this challenge by first discussing the overall goals pursued in an HRA. Then we will present a semi-formal model of essential elements and map the goals and the respective responsibility onto these elements.

The fundamental task of a safety engineer performing an HRA, is to build a model of possible interactions of the vehicle with its environment that is as close as possible to reality. Every safety engineer has such a model in his mind. Our task is to make this model explicit and guide the safety engineer in building the model efficiently. It is efficient to omit uncritical information and focus only on potentially critical parameters. Classification into uncritical and potentially critical parameters is done by reasoning logically about the interaction. This means that the safety engineer forms a concrete picture (or scenario) in his mind, with real world objects, such as a highway, ice, high-speed driving, etc. The basic argumentation of his parameter decision regarding uncritical and potentially critically parameters is done purely logically. If the safety engineer has a concrete scenario in mind that could lead to an accident (an argumentation), he needs proof for his argumentation. The proof needs to be something objective and provable. So he translates the envisioned scenario into the physical world. By physical world we mean the scientific physics[1]. While "ice" is something in the physical reality, it is not specific or precise enough for our purpose. It is unclear whether one implies with "ice" the temperature being $< 0°C$, or the friction being low $\mu < \epsilon$. Thus, for us "ice" would be a logical parameter and the concrete physical value the manifestation in the physical world. If a safety engineer argues with "ice", he might use the "evidence" of low friction $\mu < \epsilon$, for example. We can therefore state that finding a set of possibly critical scenarios is efficiently done in the logical domain of the interacting behavior of our three core elements. The amount of risk is determined by deriving physical values from the hypothesized accident. We can separate the logical domain from the physical domain by its purpose. The logical domain is mostly used for arguments and reasoning, while the physical domain is used to produce evidences for these arguments.

If we want to formalize HRAs, we need to integrate the two worlds, the logical domain and the physical domain.
Parnas' Four-Variable-Model (FVM) constitutes a (semi-) formal approach

[1] In German, we distinguish between "physisch" und "physikalisch". With scientific physics we refer to the German term "physikalisch".

for (software) system representations that makes exactly this distinction (as discussed in chapter 2.4.1 "Correctness of situation analysis"). Parnas states that the system "perceives" and "affects" the environment only physically. The behavior and strategy of the system is modeled in the logical domain. A shortcoming of using this model straightaway for HRAs is that Parnas considers everything outside the system as purely physical. Furthermore, he respects the behavior taking place "only" by one relation (called *NAT*). The core elements we need to look at are not only the system, but also extend to the human and the environment as well. We therefore extended Parnas' FVM into interacting FVMs, where each element, "System", "Environment", and "Human", is represented by one FVM. Figure 3.5 shows the interacting FVMs schematically.

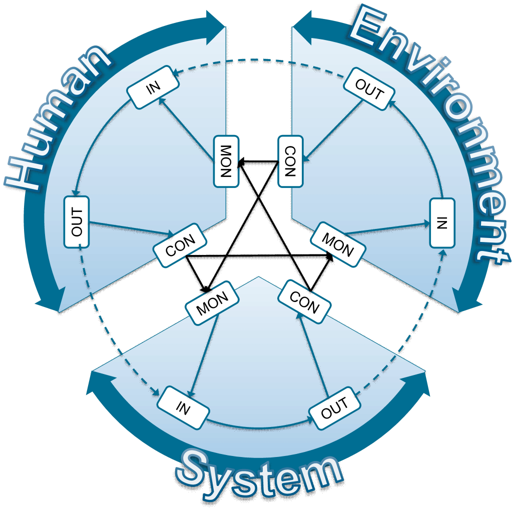


Figure 3.5: GOBI's interacting Four-Variable Models

In the figure, the physical domain is represented as the inner circle with "$MON \rightarrow CON$", while the strategic, logical world is represented on the outside circle with "$IN \rightarrow OUT$".

The semi-formal model GOBI-FVM, implemented as interacting FVMs, allows us to express the difference between figurative, logical reasoning and physically proven facts. In the following, we will refer to the corresponding FVMs by sub-scripting them:

- $\mathcal{FVM}_{Sys}$ represents the FVM of the System
- $\mathcal{FVM}_{Hum}$ represents the FVM of the Human
- $\mathcal{FVM}_{Env}$ represents the FVM of the Environment

Using this conceptual model, one can think of an HRA as being an analysis of concrete interaction scenarios of this model.
The *hazard analysis* sets up a very concrete picture of possibly critical interactions. The hazard is becoming obviously a misbehavior (service failures) of the $\mathcal{FVM}_{Sys}$. The operational situation is described by influences and characteristics of all FVMs. This step is complete if the setup of the model would inevitably result in harm of the human, if executed. We have however to add that it leads inevitably to harm, if the human does not perform any controllability action. If there exists an execution starting from the respective setup which does not end in a harmful event, the setup is not yet valid. If we assume to have a valid setup, the next step, the *risk assessment*, can be started. This step is can be subdivided into two aspects. The first aspect is that the model is executed infinitely often and all traces of executions are captured. The second aspect, the original task of the risk assessment, is to analyze these traced. The difference of various executions can only stem from actions from the human, the controllability. Everything else stays the same (by definition).

- The likelihood of controllability can be interpreted as the fraction of numbers of executions that lead to harm and the overall number of executions.
- The severity is being evaluated for the execution in which the human does not perform any controllability action.
- The exposure is the likelihood to have the respective setup, but without the hazard of the system.

The values are finally combined to an ASIL value and one hazardous event has been analyzed.

Please note that, as stated above, this is only a conceptual model to bridge the two worlds of HRAs and systems engineering. In the following we will discuss each FVM in more detail and point out general properties and peculiarities of the respective FVMs.

3.2.2 The Systems – Detailing the core elements

In this section, we will further discuss the single FVMs of the System, the Human, and the Environment.

The System-FVM

Representing a "system" as an FVM is the original purpose of an FVM. It is therefore easy to understand the sets and relations of the "System" and we will not discuss those in detail. However, we will discuss the role of the System-FVM in an HRA in the next section.

In our conceptual representation of an HRA as executions of our the interacting FVMs, the system had two roles.

The first role was to provide a misbehavior. This misbehavior is the hazard. Please note that we did not restrict the misbehavior. Thus, we include by this all possible misbehaviors, thus, we cover the case of single service failures as well as multiple service failures. The process and the risk assessment stays the same. We therefore state that the risk assessment of a hazard consisting of a multiple service failure is the same as the one for a single service failure. The differentiation whether we consider a single or multiple service failure is purely defined system-internally.

The second role the was to provide characteristics of the situations setup. It is however not defined, which characteristic might be relevant for an HRA. We answer this question as part of our OASIS discussion.

The Human-FVM

In contrast to the System, the representation of human(s) as an FVM is a new application of FVMs. Based on Figure 3.5, we define the sets compliant with Parnas' FVM (informally) as:

	Physical domain - Var_{real}
MON:	This is is real-world (physical) information humans sense.
CON:	These are real-world (physical) actions humans perform.
	Logical domain - Var_{logic}
IN:	These are concepts or ideas humans have about their current situation.
OUT:	These are intentions or plans humans have to do something.

We again omit the names of the relations between these sets and discuss their meaning directly:

$MON \rightarrow IN$: This process is often referred to as characteristics formation, or recognition. It maps real physical, sensed (raw) information to known concepts and ideas of the human.

$OUT \rightarrow CON$: This process is often referred to as concrete action. It realizes an idea of the human in terms of externally perceivable physical phenomena.

$IN \rightarrow OUT$: This is the behavior of humans. Thus, based on the ideas and their mental picture of the current situation, they will create a plan or an intention to do something.

As we have seen in our conceptual FVM-based HRA, the human has two roles in an HRA. The first role is that the human is the one who becomes harmed in case of an accident. Thus, a severity assessment must be possible purely on information contained in the human-FVM. The other role the human play is to be able to influence the interaction once, the execution has been started.

Both, the severity and controllability is not subject of this thesis. We want, however, point our the versatility of our conceptual model as interacting FVMs by presenting a possible refinement of the human FVM for the controllability part.
The controllability describes an intentional, behavioral part of the human, representing (re-)actions he can perform. We want to further refine the notion of intention, perception and action in more detail. To do so, we use a well-known model of the human behavior: the skill-rule-knowledge model of Rasmussen [Ras82]. It was created as a taxonomy for human errors. The fundamental idea is to classify human behavior by the familiarity with the task [Ras82]. Rasmussen defines that there is a tripartite distinction of task performance:

- Skill-based level: Tasks the operator is extremely familiar with (e.g., steer a car)
- Rule-based level: Tasks which are familiar, but not intuitively accomplishable (e.g., park a car)
- Knowledge-based level: Tasks that are unfamiliar (e.g., navigate through a new situation)

Figure 3.6 shows these different layers of Rasmussen's SRK model. We will go through this model and explain how Rasmussen's model can be integrated into our FVM approach. *Characteristics formation*
The purpose of the characteristics formation is processing physical inputs and transform these into logical concepts of the human. We see this as the first stage of an argument for actions and reactions or even as an argument for the uncontrollability of a situation.

This fits to Parnas' separation of the physical world and the logical world. In fact the idea of Rasmussen's characteristics formation is essentially the

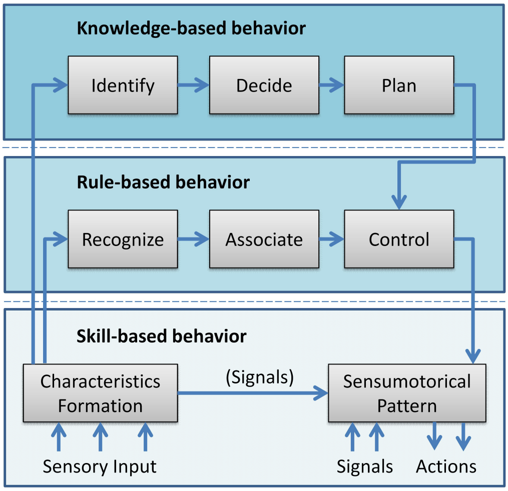


Figure 3.6: Rasmussen's Skill-Rule-Knowledge Model [Ras82]

definition of the relation $MON \rightarrow IN$. This essentially means stating the assumed model of the situation the human has in his mind.

Skill-based behavior
Informally speaking, the skill-based behavior is given by intuition and reflex-like actions. There is no conscious process involved. The behavior sequence is therefore very short and direct, based on a statement of intention. We therefore use this model as output interface putting intention into reality. It is the translation instance for transforming logical actions and reactions into physical reality, i.e., from $OUT \rightarrow CON$. This translation can define a one-to-one mapping, but is often multi-variable in the outputs, meaning that one intention usually leads to one primary physical action, but is often accompanied by secondary, subconscious actions.
If, for example, the intention is to "accelerate", this is translated into the intuitive (primary) action of moving the right foot to the gas pedal and pressing on it. Other more subtle actions (the secondary actions) are that at the same time the driver changes the focus of his view into the direction into which he expects the vehicle to accelerate. Furthermore, if the acceleration intention is high, another secondary action could be to increase the force of the grip holding the steering wheel.

Rule-based behavior
In rule-based behavior, "higher-grade" activities take place. They are of an "if-then" format, i.e. they constitute fixed, stored rules or procedure [Dav03]. Rasmussen describes rule-based behavior as "goal-oriented but structured by 'feed-forward control' through a stored rule" [Ras82][Rea90]. One may argue that the distinction to skill-based behavior is fuzzy because intention/action pairs can be represented in "if-then" format as well. The difference is the goal orientation and the consciousness of rule-based behavior.

Rule-based behavior means for HRAs that, on the one hand, we find normal maneuvers as rule-based behaviors and that controllability actions are usually rule-based as well. A controllability argument is done via the percentage of average operators that can "control" the situation. Thus, a controllability hypothesis can be stated as rule-based behavior. The task the safety engineer has to fulfill in order to argue with this rule is to show statistics on how many average drivers have the rule "stored".
The rule defines the goal, the perception and the skill-based action define human performance. Hollnagel therefore classified human error roughly into these two cases: The rule is inadequate, or the rule is satisfactory, but the performance is deficient [Hol93]. For an HRA this means that a controllability action can fail because it is inadequate or it is adequate, but the human performance is deficient.

Knowledge-based behavior
Knowledge-based behavior is similar to rule-based behavior; it is goal-oriented as well. The difference, however, is that the human states the goal very explicitly in his mind and derives a strategy or a plan for reaching this goal. Different possibilities are weighted against the best path to the goal. The result of a certain intention in terms of actions is reflected mentally. Thus, one could compare this with a mental control loop. In contrast to the mental control loop, skill-based behavior is controlled via the environment, meaning a rule is selected and an action is performed. Based on the "reaction" of the environment, the rule is adapted or another rule might be selected.

As already stated in the discussion of the skill-based behavior automaton section, our goal is to improve HRAs for automotive systems. We therefore focus the modeling of the human on controllability actions. At the time of this writing, there is no need to model knowledge-based behavior because controllability actions usually take place in a very short time, with only limited conscious influence. Strategic planning of actions in case of a system failure is more important in other domains where the human has minutes or hours of reaction time to prevent a situation from resulting in harm.

As a conclusion of this controllability refinement of the human-FVM, we want to point out the FVM idea again: the differentiation between the physical world and the logical world. Rasmussen's SRK model fits perfectly into this concept because the characteristics formation on the one hand and the sensumotorical patterns on the other hand, are exactly what the human FVM describes when transitioning form the physical world to the logical world. This integration was not meant to significantly advance the state of the art in controllability models, but we used it to show-case the very generic conceptual framework GOBI provides, is a versatile and open model.

The Environment-FVM

As discussed before, in Parnas' original FVM model, the environment was "only" represented as one relation denoted as "NAT" from the set CON to MON: $CON \xrightarrow{NAT} MON$

Due to the importance of the environment in an HRA, we need a more explicit representation of the environment. We accounted for that by representing the environment as an FVM as well. At first glance, this seems to be a straightforward extension, because the environment was already represented in the original FVM. Upon closer examination, the question arises what the logical part of the environment is? Furthermore, what are logical inputs and outputs?

The basic idea of having a physical as well as a logical domain even for the environment is that in an HRA, we often find informal conceptual properties of the environment. The physically *relevant* properties are often implicit, and which aspect is relevant may change dependent on the expert and the hazard. One example could be "ice" on the road. On the one hand, "ice" is something physical and one could expect to find this in the physical domain. For an HRA, however, "ice" is only a logical representative of something physically more concrete/measurable, such as the temperature, or the friction of the ground (μ). We therefore distinguish between influences that are semantically meaningful for describing a concrete situation and influences stating values or properties of technical, physical quiddity.

In the following, we will detail the FVM as we did with the other two GOBI-FVMs before. Based on Figure 3.5, we define the sets compliant with Parnas' FVM (informally) as:

	Physical domain - Var_{real}
MON:	These are real-world (physical) values that are put into the environment.
CON:	These are real-world (physical) values that are created by the environment.
	Logical domain - Var_{logic}
IN:	This set represents known relevant influences or influence classes.
OUT:	This set represents suggested influence factors.

The meaning and purpose of these sets alone is quite fuzzy and hard to grasp. Their nature becomes clearer once we discuss their relations:

$MON \rightarrow IN$: The MON set contains values and variables present and therefore known in the physical world. Dependent on these physical values, the environment can be constrained into "incredible" and possibly "credible" situation facts. If the "System", for example, provides the information that the hazard is "loss of steering", the subset of situations where no steering is required, "steering angle = 0", is irrelevant for the HRA.

$IN \rightarrow OUT$: This relation represents formalized domain knowledge in terms of situation influence dependencies. With domain knowledge we can sub-classify the remaining set of possibly "credible" situations, or at least partition the remaining state variables into "equivalence" classes.

$OUT \rightarrow CON$: This relation finally makes the domain-knowledge explicit in the physical domain. Thus, if we identified a partition of logical environmental influences, these classes need to be translated into reality, i.e., we need to define corresponding physical variable partitions.

We can therefore interpret the environmental FVM as a three-step process:

$$\overbrace{Analysis}^{MON \rightarrow IN} \rightarrow \overbrace{Reasoning}^{IN \rightarrow OUT} \rightarrow \overbrace{Synthesis}^{OUT \rightarrow CON} \tag{3.1}$$

- The analysis part restricts the logical values based on incoming physical values. If we know, for example, that the vehicle speed is $v_x > 100km/h$, the analysis would produce the knowledge that the situation cannot be a "parking" nor a "city-driving" scenario.
- The synthesis part does the opposite: restrict the physical world based on logical values. If the synthesis gets as input "ice", it concludes the restriction that the temperature "$t < 5°C$" and "$\mu = low$". Another synthesis would be that "parking" implies "$v_x = 0$".
- The reasoning part finally "decides" on dependencies of logical influence factors and suggests possibly relevant other influence factors (logically).

Let's get back to the steering system and assume a "self-steering" hazard. The vehicle can only state actor-related values, such as the steering angle $sa > 0$. The analysis part partitions the logical facts into irrelevant and relevant facts. "Parking" would belong to the irrelevant set, while "driving straight" would belong to the relevant influence factors: If the vehicle is "parked", nothing happens if the steering angle changes. However, if one wants to "drive straight" and the vehicle (suddenly) starts to turn, the situation gets critical.

The analysis part "communicates" this to the reasoning part. Based on "driving straight", the reasoning part could conclude that, for example,

"Country Road" or "Highway" are important logical situation facts relevant when "driving straight". Additionally, "upcoming traffic" or "heavy traffic" could be relevant.
The synthesis part translates the suggestions of the reasoning into physically relevant partitions of variable valuations. From "Country Road" and "Highway", the synthesis can produce, for example, classes of vehicle velocities, such as $50km/h \leq v_x \leq 100km/h$. Another example is that from "upcoming traffic", one could conclude a high differential speed: $\Delta v_x \geq 50km/h$.

We can state that the environment-FVM takes over a huge part of the challenges identified for the formalization of operational situations. While everything in the environment seems to be physical, our notion of being physical becomes evident with this refined FVM: It is semantical preciseness. Additionally to the preciseness the environmental model implies, the reason part is, as discussed above, one fundamental aspect of domain-knowledge, we want to formalize situation definitions and for defining exposure assessment consistency.

3.2.3 GOBI conclusion

In GOBI we presented a semi-formal model of interacting Four-Variable-Models. Furthermore, we mapped the general HRA process of *hazard analysis* and *risk assessment* to that model. By this, we derived basic requirements for the remaining core challenge, which will be discussed in the next three sections. We handled the challenge to capture the fuzzy concept of an HRA in an extended version of a commonly known model, a FVM. Our contribution with GOBI is therefore the idea of representing and formalizing an HRA conceptually as interacting Four-Variable-Model.

3.3 OASIS - Ontology-based Analysis of Situation Influences on Safety

In the previous section, GOBI, we presented the foundation of this thesis; a model-based HRA representation. In this section, we address the first part of the challenge of providing a more structured and consistent approach of HRAs in general. In this section, we focus on the formalization of operational situations. Figure 3.7 shows the integration of OASIS with the overall thesis on the one hand and the general structure of this section on the other hand.

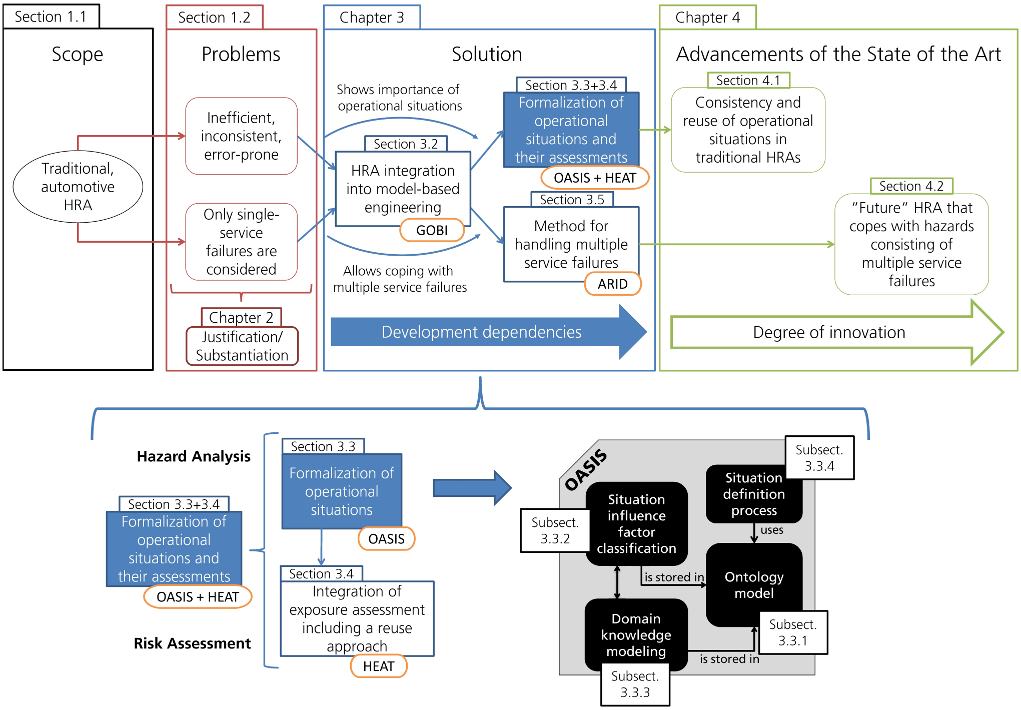

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method for handling multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation
Hazard Analysis
Risk Assessment
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.3
Formalization of operational situations
OASIS
Section 3.4
Integration of exposure assessment including a reuse approach
HEAT
OASIS
Subsect. 3.3.2
Situation influence factor classification
Situation definition process
Subsect. 3.3.4
uses
is stored in
Ontology model
Subsect. 3.3.1
Domain knowledge modeling
is stored in
Subsect. 3.3.3

Figure 3.7: Integration overview of OASIS

In section "2.2 Core Challenges and Goals", we identified a structure of relevant challenges. The following branch was one of the paths of derivation of concrete challenges on the way to a formalized, model-based HRA:

- ✧ Structuring of hazard analysis
  - ✧ Correctness of situation analysis
    - ➠ Formalization of operational situations

Additionally, we derived as part of the work related to "correctness of situation analysis" (cf. section 2.4.1), we derived the following findings, which we will transfer to the application of HRA-formalization in this section.

1. Situation theory uses infons, which are relations containing information items about a situation. In situation theory, the relations and relevant information items to consider need to be adapted for each purpose. Thus, in our case we need to define this for HRA situations.
2. Based on the context modeling survey of Strang and Linnhoff-Popien, we argue that ontological models are the most suitable type of model for capturing HRA situation information.

Thus, in this section we will present our solution for formalizing environmental information for the purpose of being used in an HRA. This challenge can be subdivided into the following structure, which directly leads to the structure of this section.

1. Strang and Linhoff-Popien claimed that ontological modeling is the most suitable type of model for situation/context modeling. Nevertheless, due to the observation that the term "ontology" is used for many different things, we start by giving an overview of different ontology applications and introduce concrete names for these applications. This helps to focus on the various aspects an ontology can express and makes it easier to define the relation between an ontology and other similar concepts such as taxonomies and meta-models. This will be done in section 3.3.1.
2. As mentioned above, we need to define "infons" for HRAs, i.e., we need to adapt situation theory insights to HRAs. Up to now, we have not defined how to structure "influence factors" of HRA situations. Thus, we need to answer the questions how the "environment" can be structured and what actual information items could look like. As a concrete technical solution, we use an ontology as a model. We will discuss all these issues in section 3.3.2.
3. Knowing the building blocks of an "environment" (in terms of HRAs) is necessary, but this does not include any information about what a reasonable situation is. If the task in situation theory would be to build an HRA situation "world", one would ask for knowledge in order to decide whether some infons make a situation factual or not (cf. section 2.4). In our case, the HRA, this translates to the question of whether a combination of influence factors makes sense or not, or to be more precise, how likely the combination is. How to capture this domain knowledge will be the subject of section 3.3.3.
4. The overall purpose of this section is to come up with a more formal way of defining situations for an HRA. The structure introduced above gives us the (formal) basis for defining situations in general. Another aspect is how to get relevant/meaningful situations. We will therefore define a situation definition process in section 3.3.4.

After reading this section, the reader will know how to model situations for an HRA by using our formalization of (HRA-relevant) environmental information.

3.3.1 Ontologies

As described above, the term ontology is used for many different application classes. These application classes include glossaries or data dictionaries, taxonomies and thesauri, schemes and data models, and formal ontologies for inference. Additionally, one can often read discussions about ontology vs. meta-models. All of these concepts are, of course, not independent, but they are distinguishable.

Our goal in this section is to explain the ontology characteristics for usage in an HRA situation modeling scenario. We will therefore first give a general overview of different ontology application classes and differentiate them. Afterwards, we will apply these concepts to our goal of formalizing HRA situations.

Ontology applications

One commonality of all ontology applications is the intention to "capture" a certain world. The "world" is the ontology term for a set of concepts and instances contained under a common "umbrella". The amount of structure this "umbrella" produces is our differentiating element in the discussion.

The first level of structure is the explicit enumeration of terms belonging to the "world" under consideration. This can be roughly defined as *vocabulary*. If there is an authority ensuring the unambiguity and non-redundancy of terms, we have the second level of structure, a *controlled vocabulary*.

The next level of structure is introduced by turning a *controlled vocabulary* into a *taxonomy*. A *taxonomy* provides not only well-defined terms, but organizes the terms into a hierarchical structure. Each term in a *taxonomy* is in at least one parent-child relationship. Traditionally, this parent-child relationship has been a generalization/specialization or a "is a kind of" relation. Nowadays, one can often see other meanings of hierarchy, such as "part-of" or "instance of".

If the relationship is not limited to hierarchy, but adds associative relations between terms, we speak of a *thesaurus*. Other relations known from a word-based thesaurus are synonym or antonym words.

The step from a *thesaurus* to an *ontology* is the definition of a grammar. An ontology grammar states what a valid "world" looks like, i.e., what is meaningful/expedient within a specified domain of interest.

Last but not least, we have the term *meta-model*. The focus of a *meta-model* is to construct specific models within a domain of interest. To this end, a *meta-model* is an explicit model of the constructs and rules needed to specify correct models. *Ontologies* in contrast, do not necessarily define the grammar such that it can be used as a model construction plan. The distinction is minor in concepts, but often huge in practice. The key is the "explicity" of meta-models wrt. construction rules. Thus, a *meta-model* is an *ontology* (that can be used by modelers), but not every *ontology* is explicitly modeled as a *meta-model*.

The following figure summarizes the above review of ontology applications graphically.

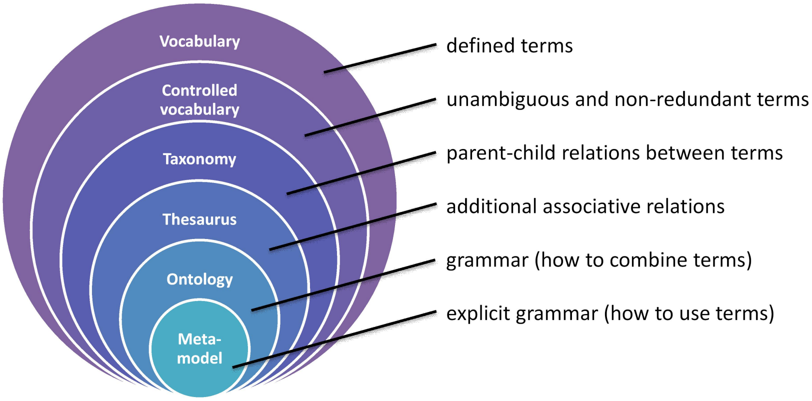

Vocabulary
Controlled vocabulary
Taxonomy
Thesaurus
Ontology
Meta-model
defined terms
unambiguous and non-redundant terms
parent-child relations between terms
additional associative relations
grammar (how to combine terms)
explicit grammar (how to use terms)

Figure 3.8: Taxonomy of ontology applications

Ontology requirements for HRA situations

In OASIS, we want to define situations in a more formal and explicit way. The model of a situation is very simple: It is a list of influence factors. Thus, our *vocabulary* contains mostly influence factors. Due to our goal of reusing situations, we need *unambiguity* of influence factors; otherwise, a situation (and its assigned exposure value) might be reused in a setting for which the situation was not defined. *Non-redundancy* would be appealing, but by demanding it we would have to sacrifice "practicability". This is true because engineers from different domains speak (often slightly) different languages in the sense that they use different terms for similar aspects. One example could be "*ice*" and "*low* μ"; both terms imply that the friction of the road is reduced. "*ice*" is a more figurative term, while "*low* μ" is a technical, physical expression. We would like to respect this (practicably implied) scenario on the one hand, but do not want to have arbitrary redundancy in our ontology on the other hand. Therefore, we allow redundancy but offer a way to "unify" redundant terms. Unification means in this case that a situation would be identified as similar if only the redundant terms were switched. The approach for unification will be explained in section 3.3.3

Setting up the influence factor structure as a taxonomy, i.e., including hierarchical concepts, has multiple advantages for the definition of HRA situation influence factors:

1. *Handling of complexity*

   Hierarchies are a way of handling complexity and structuring huge sets of information. A vehicle's environment is a complex "set of information".

Thus, due to the goal of formalizing this in terms of influence factors, some structure and organization is necessary.

2. *Keeping consistency*

   Separation into classes (or hierarchical structures) has the advantage that it becomes easier to decide about the attributes introduced above: unambiguity and non-redundancy. If each level of the hierarchy is unambiguous and non-redundant (and if the influence factors are assigned to the correct classes), we do not need to check each and every pair of influence factors, but can already restrict the analysis to the class level. This reduces the effort for maintaining a consistent ontology.

3. *Support influence factor retrieval*

   For *defining* an HRA situation, it is important to find the influence factors the engineer has in mind. Thus, an HRA influence factor model needs to support the retrieval/finding of influence factors. A hierarchical structure is easier to browse.

With the above-stated requirements for an HRA taxonomy, we have good requirements for handling the single influence factors. How these influence factors are related to each other and what their "meaning" is for an HRA situation has not been discussed yet. As we have already stated as part of the "non-redundancy" discussion above, we need at least one associative relation defining a unification property for relating two influence factors to each other. Furthermore, the domain knowledge of safety experts in HRA encompasses more aspects that define further relations between influence factors. For our goal of formalizing HRA situations, this domain knowledge is very valuable. There are two reasons for this:

1. *Decide about validity of situations*

   We need domain knowledge to decide about the validity of a situation. (This in turn helps us not to introduce useless situations in our database and keep it practicably usable.)

2. *Support consistent exposure assessments*

   Domain knowledge must be formalized in order to achieve our goal for supporting the engineer in consistent situation assessments. Consistency is laxly defined as fairness of situation assessments. To decide on whether something is fair, one usually needs a notion of similarity. For deciding whether situations are similar or not we need domain knowledge.

Hence, transitioning from an influence factor organization model to a situation-focused one is a step from a taxonomy to (at least) a thesaurus.

The relations are also an essential part of our (implicit) grammar. Please remember that a grammar turns a thesaurus into an ontology by defining *what* correct combinations of concepts and instances are. This does not necessarily include *how* to construct a useful combination. For us, the handling and the decision regarding the expedience of a situation is the main focus (not the modeling of situations). Therefore, for our approach, an *ontology* is appropriate/sufficient.

A meta-model, in contrast, adds the aspect of explicit modeling, thus the constructive aspect. We do not see a huge challenge to "model" a situation (as a list of influence factors). Therefore, there is no need for a meta-model-based approach.

Note that we will introduce a meta-model and a modeling approach in section 3.3.3, but this meta-model does not describe the explicit modeling of situations, but rather domain-knowledge modeling of influence factors.

Based on the structured review of ontology applications shown above we structure the subsequent chapter as follows:

First, we will introduce an HRA taxonomy by classifying and defining situation influence factor classes.

Second, we will turn the taxonomy into an ontology by defining rules or relations for capturing domain knowledge for HRA situation definitions.

3.3.2 Situation influence characterization - a taxonomy

In this section, we present our HRA influence factor taxonomy. The taxonomy is extracted from industrial case studies and project experience. On the one hand, we respect the fact that an environment description can always be extended, detailed, or refined, meaning there no *the* final HRA environmental model; on the other hand, one can define a fixed (non-extensible) general structure on which an extension can be based.

Thus, we claim that the first two levels of our taxonomy form a complete and correct basis for HRA influence factor classification, while the more detailed classes and concrete influence factors are an initial, but extensible influence factor model. Please remember our discussion as part of the related work section 2.4 on "***Efficient reuse of situations***", where we discussed that trying to build a complete model for HRA influence factor models is not practical.

First-level OASIS taxonomy

A good practice for engineering taxonomies is to have one common "parent" for all other terms. Our root or top-parent is "OASIS Characteristic". This means that everything defined as a child of OASIS Characteristic is an influence factor of OASIS.
The second level of hierarchy can be adopted from GOBI (cf. Chapter 3.2). In GOBI, we discussed that we have three participants in an HRA: The system or vehicle, the environment, and the human. Therefore, we defined as a first level of our taxonomy these participants as concepts for this level. In other words, if we are talking about a certain HRA situation, we can subdivide the single facts into contributing characteristics from the vehicle, the environment, and the human. For the latter we extended or focused the classification more on the fact that humans act or interact with the system, while the environment is "just there". The taxonomy concepts are termed accordingly: "***VehicleCharacteristic***", "***EnvironmentalCharacteristic***", and "***InteractionCharacteristic***".

Separating these characteristic classes along our GOBI model gives us the unambiguity and non-redundancy property for these classes because an influence factor can describe either the vehicle, the environment, or an interaction aspect. If we take the original FVMs of GOBI, namely system, environment, and human, this strict separation becomes evident.

More interesting is the refinement of the characteristic classes introduced above. This is the second level of our taxonomy, which we will discuss in the next section. Figure 3.9 shows the OASIS root and the first two levels of the characteristic hierarchy.

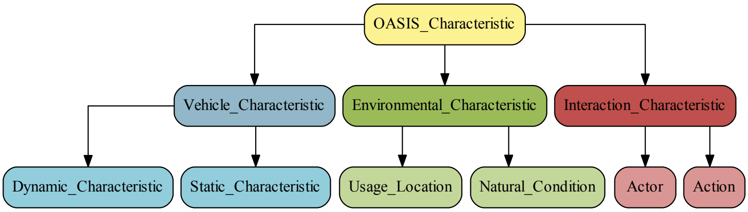


Figure 3.9: OASIS' top levels of influence factors

Second-level OASIS taxonomy

In the following section, we will discuss the second level of our situation influence taxonomy. Additionally, we will show the specific instances we identified during our work. Note that we claim that the hierarchy is correct and complete, but we do not claim the concrete model to be complete. This model is subject to continued learning and extension.

VehicleCharacteristics are all situation influences that describe the state of the vehicle in more detail. Examples are the vehicle's velocity, its acceleration, "clamp 15" on, or having a roof luggage rack installed. For an HRA it is essential to know how vehicle states can change. Each change of a vehicle state introduces a possibility to switch to a bad situation (which we need to be aware of). A vehicle can change its state on duty or off duty. With "duty" we mean the intended usage of a system: in our case, driving the car. Due to the fact that these are easily separable state changes, we subclassified "VehicleCharacteristic" into two concepts: *static* and *dynamic characteristics*.

Static characteristics are those that are persistent at least throughout one driving cycle and change at most off duty. They describe the setup or configuration of the car in more detail. The setup itself might change the general physics of the vehicle, such as wind resistance, weight, gravity center, in a way that affects the behavior of the vehicle (on duty). "*Roof luggage rack*" or "*trailor attached*" are examples of static characteristics. Both change the general physics of a vehicle and need to be considered in an HRA. Thus, we use this class to describe general setups in the sense of persistent structural or functional features. In our industrial case studies, influence factors in this class were most of the time mentioned as constraints in the item definition and not explicitly mentioned in the situation description. The item (should), however, describes a functionality and not a physical structure. We therefore moved this information into our situation ontology as a static characteristic.

We did not identify any further sub-classification in this class. Furthermore, only eight concrete influence factors were extracted.

Dynamic characteristics, in contrast, contain influence factors for describing the vigorous nature of driving situations, such as the vehicle's velocity and its current acceleration. Thus, dynamic characteristics may change fast and often during one driving cycle, i.e., on duty. Additionally, there are dynamics included that may change but do not change as often and fast as the vehicle's velocity. These describe concrete states, such as "drivers seatbelt buckled up" or "charging cable connected" (in case of electrically driven cars). Therefore, one can think of dynamic characteristics as descriptions of states on the one hand (if discrete) and the definition of concrete physical values on the other hand (if continuous).

In the following, we will give some rationales for the next (third) level of refinement of dynamic characteristics.

✓ *Vehicle State*

This is a container class containing state based vehicle influence factors. This class includes, for example, the influence factor "Interior_Open", referring to the scenario that we describe an opened convertible. Other example influence factors are "vehicle_locked" or "battery attached". All optional "features" described as static characteristics have their concrete states listed as influence factors of this sub-class.

✓ *Delta Speed*

With this class we can specify the difference in speed between the vehicle under consideration and other traffic participants. We could theoretically distinguish lateral and longitudinal delta speeds, but in practice we have not found a case in which an explicit lateral delta speed was mentioned.

✓ *Distance*

Besides the difference in velocity, the distance to an obstacle or another traffic participant is crucial for deciding on the failure tolerance time.

✓ *Velocity lateral*

The lateral velocity is often referred to in highly dynamic situations such as driving in a curve.

✓ *Velocity longitudinal*

The longitudinal velocity is obviously important.

✓ *Acceleration lateral*

Besides the lateral velocity, the lateral acceleration is often used in vehicle dynamics to reason about the stability of the vehicle.

✓ *Acceleration longitudinal*

The longitudinal acceleration is obviously important.

✓ *Steering Angle*

The steering angle expresses the drivers wish to drive a curve. It can be found in many situation descriptions of HRAs, but one has to be careful not to confuse the steering angle with the curve radius of the street. The latter describes the actual path the vehicle should take, whereas the steering angle describes only the driver's projection and might differ from the actual path driven (due to drifting).

The above classification was evaluated in industrial case studies and automotive projects and a list of initial subclasses and a concrete set of influence factors were introduced. While the separation into "on and off duty" introduces unambiguity and non-redundancy in theory, we experienced that in practice there are still borderline cases. The snapshot shown as Figure 3.10 shows the subclass "*Vehicle State*" as a dynamic characteristic. This class includes, for example, the state of "*Clamp 15*". Clamp 15 refers to the state of whether the car is operational or not (kind of "ignition"). Thus, one could argue that if clamp 15 is off, the car is by definition in off-duty mode. While this is true, it is still misleading because a parked vehicle is "on duty" as well: The brakes have to hold the car! Clamp 15 is therefore dynamic throughout the vehicle's duty cycle. The other differentiating aspect is the physical implication. As discussed above, static characteristics change the general setup and the physical configuration of the car. Dynamic characteristics might express physical characteristics as well, but these are situation-dependent physical values, such as *Acceleration, Velocity, Delta Speed*.

Another lesson learned is that we need to distinguish features and their state or usage. One example is the "feature" of being a "convertible". Being a convertible is a static characteristic because it defines a general vehicle setup. The state of being "converted" in the sense that the roof is open is, however, a dynamic characteristic! In this sense we could include "Clamp15" as a feature in the static characteristics and keep the influence factors clamp 15 on/off in the dynamic characteristics. Due to the fact that every car has a "Clamp15", it does not really make sense to include this in the static characteristics (it would be included in all situation descriptions!).

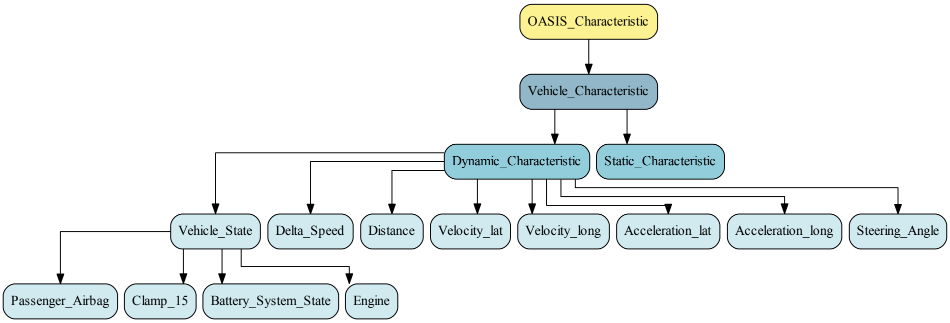


Figure 3.10: OASIS' Vehicle Characteristics classes

EnvironmentalCharacteristics contain all situation influences that are part of the context of the vehicle. Thus, the environmental characteristics describe the situation surrounding the vehicle in more detail. Thus, if one wants to express a driving situation such as "*approaching a congestion on a highway in winter with ice*", the context information are: "*congestion*", "*highway*", "*winter*", and "*ice*". One important aspect of HRA situations is their likelihood. To determine the likelihood of a situation, we reason about all influence factors and their dependent likelihood. Some influence factors are more likely if we already included others. If we have "*driving at 150km/h*" already, it is very likely that we are driving on a highway, for example. During our analysis of situations of industrial case studies, we discovered, however, that some (environmental) influence factors are not dependent on any other influence factor classes. The class we are talking about are natural phenomena. We separated these influence classes into their own class "Natural Condition". The advantage of doing this is that, by definition, all influence factors in this class are independent (from other classes). Note that these influence factors can, of course, depend on themselves. The advantage of being independent lies in the situation assessment. If we take, for example, the influence factor "ice" and say that ice occurs during around 2-3% of the year, we could derive an exposure reduction of 2 orders of magnitude (roughly). Due to the independence we could now take every other situation Sit with the respective exposure value, let's say E3. By adding the influence factor "ice" we can mathematically determine the exposure of the new situation, $Sit \cup \{ice\}$, being 3 - 2 = 1 (E1).

Besides "Natural Conditions", which are independent of Interaction- and Vehicle-Characteristics, there are environmental influence factors that are not independent. "Street type" or the current "traffic situation" are, for example, certainly not independent of the likelihood of the vehicle's "velocity". One commonality of all these dependent environmental influence factors is that these are all man-made and the likelihood of their usage/encounter depends on the purpose and the setup of vehicle.

The differentiation into *purely natural, independent*, and *man-made dependent* environmental influence factors makes these classes disjoint and therefore unambiguous and non-redundant. We termed the two classes "*Natural Condition*" and "*Usage Location*".

We introduced *Natural Condition*s to capture all natural phenomena. The rationale is that all purely natural things are independent of almost all other influence factors. Nature creates an effect and humans can try to compensate for the effect, but cannot avoid their occurrence. Thus, the influence factor is there, independent of other vehicle-specific influence factors. For example, one cannot stop the weather from being cold, but one can heat the vehicle. Note that one could infer some knowledge, such as if the heat is on, the probability of the weather being cold is high. This is, however, a probabilistic issue. The probability of the weather being cold is generally independent of the heating system being on! The basic natural issues described in this class are time, season, and weather. All of these cannot be influenced by humans.

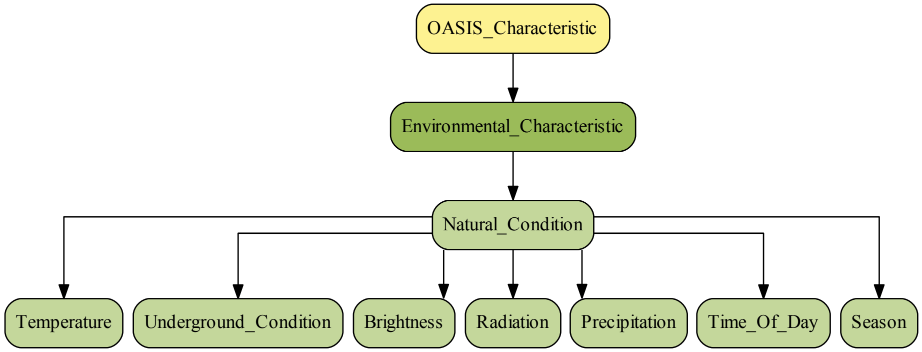


Figure 3.11: OASIS' Environmental Characteristics classes branch "Natural Condition"

In the following list, we will give some rationales for the subclasses we selected for "Natural Condition".

✓ *Temperature*

The environmental temperature is important for deriving potential hazards, such as overheating of the engine if the temperature is too hot, or reduced grip if the temperature is too low

✓ *Underground condition*

The condition of the underground is obviously important. However, as subclass of natural condition this class contains those influence factors that are influencing the underground purely natural. One possible subclass could be natural coverage, such as foliage or ice.

✓ *Brightness*

Brightness was included for two reasons: 1) is is important for the sight of the driver and 2) it is important for many driving assistance systems. Brightness was, for example, used in a case study to express glaring effects that could disturb or distract the driver or other traffic participants.

✓ *Radiation*

Radiation was mentioned only once in the case studies. It has only one influence factor "UV radiation", which was used to describe situations where camera-based systems might be disturbed. Other important radiations might be possible, such as electro-magnetic radiation.

✓ *Precipitation*

Precipitation is one of the "classic" environmental parameters. It contains everything which physically/materially falls down on the vehicle. Typical precipitations are rain, sleet, and snow.

✓ *Time of day*

Time of day was included due to two aspects: 1) The time of day implies a certain brightness and 2) it can be used to reason about the vigilance of humans.

✓ *Season*

The season was included because it describes a general setup. If we start in a "winter"-situation for example, we already have many concrete aspects in mind, such as "ice", "cold", and "dark". Stating a season does not specifically define a situation, but reduces the set and the mental picture of a situation.

For *Usage Location* we could extract the following subconcepts from the industrial case studies: "*Street type*", "*Surrounding Area*", "*Traffic Control Element*", and "*Traffic Situation*". "*Congestion*" is obviously a "*Traffic Situation*", "*Highway*" belongs to the class "*Street type*", and "*Pedestrian Crossing*" is a "*Traffic Control Element*". The class "*Surrounding Area*" does not describe the context in general, but more global aspects of a scenario, such as "*Charging Station*", "*Garage*", or "*Conservation Area*".

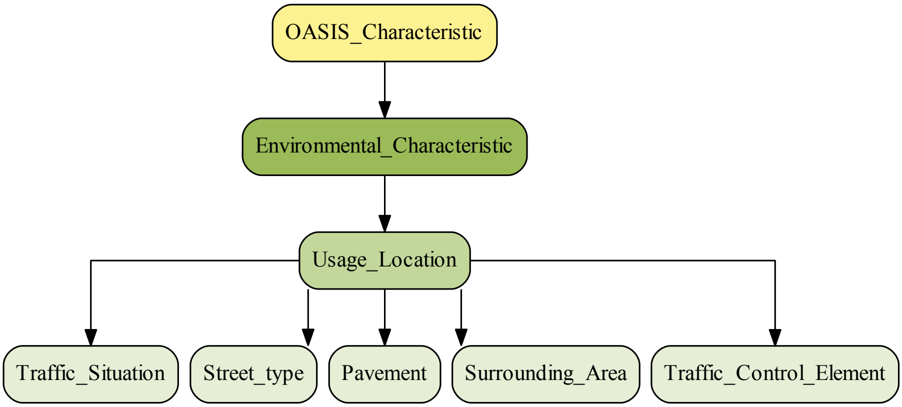


Figure 3.12: OASIS' Environmental Characteristics classes branch "Usage Location"

In the following list, we will give some rationales for the subclasses we selected for "Usage Location".

✓ *Traffic Situation*

Traffic situation is another "classic" sub-class. It describes other traffic participants and their movement. Typical influence factors are: "congestion" or "stop-and-go".

✓ *Street Type*

With the street type we describe the general class of the street. The class of a street implies many other influence factors. "Highway", is for example a influence factor in this class: By stating "Highway" we usually imply high velocity and no pedestrians.

✓ *Pavement*

The pavement is important because it describes the direct contact between the vehicle (wheels) and the environment. This significantly influences typical controllability actions, such as braking.

✓ *Surrounding Area*

The surrounding area is another class which gives us a big picture, but seldomly details. Thus, it sets up a common coarse-grained situation frame, which needs to be detailed further. Due to the fact that talking about the surrounding area is quite common, we included this class in the ontology. Typical influence factors are: "city area", "residential area", "parking lot".

✓ *Traffic Control Element*

Traffic control elements are all elements that somehow influence the flow of the traffic. A crossing or a traffic light, for example, are elements forcing a vehicle to stop (if the light is red).

InteractionCharacteristics complete the set of first-level influence factors by including the human. We named the class "*InteractionCharacteristics*" because not only humans directly, but other vehicles (driven by humans or machines) might also interact with the system as well. Thus, everyone somehow contributing active influences to a situation is in the *InteractionCharacteristic*s class. A (partial) situation such as "driver is overtaking in spite of oncoming traffic" is purely an interaction scenario containing the influence factors: "*driver*", "*overtake*", and "*oncoming traffic*".

Interactions always contain interacting participant and the kind of interaction, a kind of effect. The terms usually used are action and actors. We therefore sub-classified the *InteractionCharacteristic*s into "*Action*" and "*Actor*". These classes are obviously disjoint and therefore yield the properties of being unambiguous and non-redundant.

The class "*Action*" describes the usage of a vehicle. The expected usage of a vehicle includes, of course: "*Drive*", "*Park*", "*Accelerate*". These we termed "*Standard Usage*" because this class contains all the normally expected usage scenarios one would think of. Besides these primary use cases, however, we identified a secondary class of use-cases as part of the industrial case studies. We named it "*Service Usage*". It comprises all maintenance and repair actions one can think of.

The following list gives further rationales for the two sub-classes introduced above.

✓ *Service Usage*

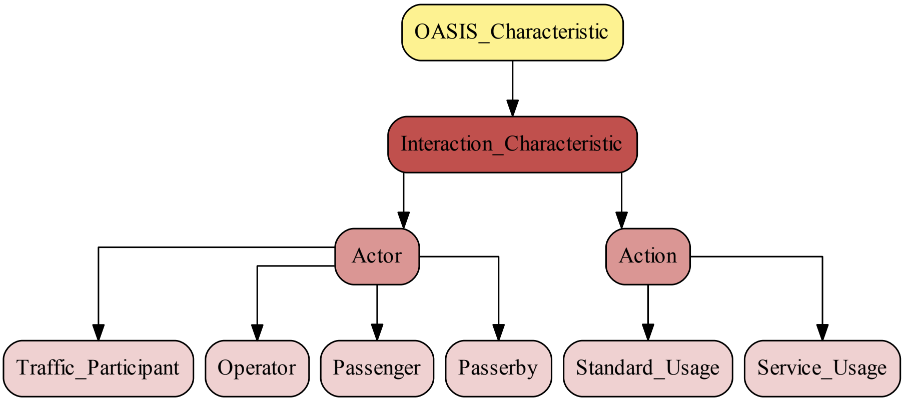


Figure 3.13: OASIS' Interaction Characteristic's classes

Service usages describe special cases of actions. Special for "safety" because one can argue with certain assumptions regarding the actors, such as trained personnel. Another "special" aspect is that there are currently discussions about whether a "workshop" scenario is covered by ISO 26262 at all.

✓ *Standard Usage*

These are the regular "maneuvers" an actor can do with a car. He can "honk", "steer", "accelerate", etc.

In terms of "*Actors*", we subdivided the class into persons who are inside the vehicle or are directly concerned with the vehicle and other persons in the proximity of the vehicle. The classes of persons who are strongly related to the vehicle are the "*Operator*"s and "*Passengers*" of the car. The operator is, of course, the first instance that directly influences the vehicle's behavior in regular situations as well as in exceptional situations, e.g., though controllability actions. The other classes of actors are "*Passerby*" and "*Traffic Participant*". Both classes are characterized by the fact that they either contribute sudden (hazardous) actions, and/or are endangered. The difference is the involvement in traffic. A passerby is not directly taking part in traffic, whereas a traffic participant (obviously) participates in traffic.

In the following, we will give some rationales for the sub-classes we defined for "Actor".

✓ *Operator*

The default operator is, of course, the "driver", but for many commission issues, the case that "no driver" is present is important as well. Additionally, we need to express the expertise of workshop personnel (if we want to argue with that). Thus, this class contains an influence factor "mechanic". Overall, this class contains all active and directly influencing actors.

✓ *Passenger*

The passengers can directly and indirectly influence the vehicle. The main difference is that they are not the main actor OR (in HRA situations) do not act as expected. One secondary vehicle action is to open a window. An example of unexpected behavior is a seatbelt being unfastened or a dog suddenly barking or a child romping.

✓ *Traffic Participant*

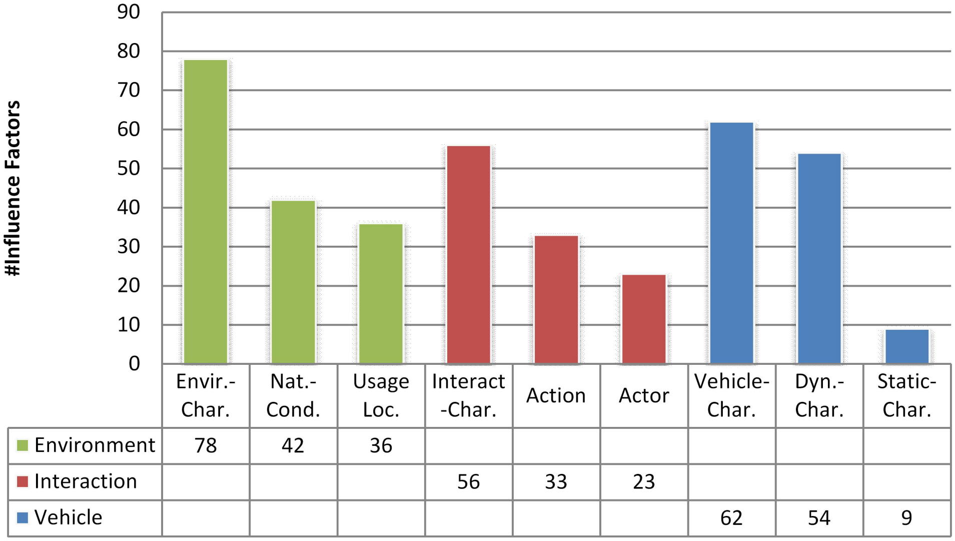
Similar to the active or direct and indirect passengers/operators, other traffic participants can be directly active or indirectly important. A traffic participant can be an obstacle to which the driver needs to react. Another possibility is that the traffic participant actively changes the braking distance by accelerating or and by performing steering actions to avoid a collision. We state that all traffic participants actively influence the vehicle or the behavior of the operator.

✓ *Passerby*

A passerby, in contrast, usually does not have a direct influence, but an indirect one by distracting the driver. A passerby could also be the victim of harm.

The third-level classes we presented for all second-level classes are based on industrial case studies of HRAs and were extracted from around 300 situation descriptions. These situation descriptions led to a total amount of around 200 concrete influence factors. We do not list every influence factor here, but would like to give an impression of the influence factor distribution into classes.

Figure 3.14 therefore shows how many concrete influence factors are in the first- and second-level classes. Note that the sum of the second-level gives the respective first-level value.



	Envir.-Char.	Nat.-Cond.	Usage Loc.	Interact-Char.	Action	Actor	Vehicle-Char.	Dyn.-Char.	Static-Char.
🟩 Environment	78	42	36						
🟥 Interaction				56	33	23			
🟦 Vehicle							62	54	9

Figure 3.14: Influence factor instances in OASIS’ ontology

3.3.3 Domain Knowledge Integration - the Ontology

In the previous chapter, we introduced our OASIS classification of influence factors as a taxonomy. In this section, we introduce the missing piece needed to turn a taxonomy into an ontology: the grammar. The grammar's purpose is to define how the vocabulary of a taxonomy can be used in a certain domain. We are in the domain of HRA situation definition and assessment and our goal is therefore to provide a means for capturing domain knowledge on the one hand, and, on the other hand use this knowledge to support the safety engineer in the task of assessing a situation's exposure.

As part of the last section, we already identified the need for a unifying relation. Additionally, we discussed the possibility of automatically determining the exposure if we have a situation and adding only an independent influence factor. If we want to achieve this for dependent influence factors, we need to know the factors of dependency. This dependency usually results in an adaptation of the exposure value. However, this is domain knowledge and needs to be captured in our model. An extreme case of this exposure value adaptation is if two influence factors are disjoint, meaning that they cannot occur together (or at least that such an occurrence is highly unlikely). All situations comprising both influence factors are obviously not very useful and can be removed from the situation knowledge base.

Thus, the fundamental challenge is to model dependencies among influence factors (which express domain knowledge) to reason about their dependent likelihood. We evaluated whether we can use Bayesian networks for calculating the resulting dependent likelihood, but due to the fact that Bayesian networks are directed and acyclic, they were not appropriate for our approach. We would have to create a directed acyclic graph of dependencies of influence factors, which does not reflect the nature of influence factors. Other approaches, like Conditional Random Fields, allow cyclic dependencies, but require complete domain knowledge. This is in our case not practical and not possible to model. Therefore, we need a more lightweight possibility to capture domain knowledge. Due to the constraint of incomplete domain knowledge, we analyzed what kind of knowledge can be expected to be entered by experts. We came to the conclusion that we can limit our model to capturing pairwise dependencies of influence factors only. The alternative would have been to model dependencies of sets of influence factors as well. Besides the observation that this knowledge is hard to get, from a practical point it is challenging to enter all the domain knowledge because the higher the order of the sets of influence factors, the more sets we can build and therefore the more dependencies we would have to define. That is why we limited ourselves to the "simple" dependency of two influence factors.

Our approach - a trade-off between theory and practice

The dependency between two influence factors influences the exposure value (as already discussed above). If we would like to formalize/capture this change in exposure, we can structure the way the influence takes place into three cases:

1. There is no change of the exposure (of a situation) if we add a dependent influence factor to the situation. In this case, the dependency describes the above mentioned unification.
2. Independent of the value of the exposure of the situation, if we add a dependent influence factor, the combined situation becomes highly unlikely. In this case, the dependency describes the disjoint property discussed above.
3. The dependency changes the exposure value of a situation. The addition of an influence factor can only make the situation more specific and thus, the change is limited to a reduction of the exposure value.

Please remember, in ISO 26262 "exposure" is one of the three HRA assessment parameters. It can hold values from E0 to E4 and each step can be thought of as an increase of one order of magnitude in likelihood (compare the state-of-the-art discussion as well).
In order to formalize the impact on the exposure value of our domain knowledge capturing relations, we defined a functional notation for exposure:

$exposure : SituationSet \longrightarrow \{E0, E1, E2, E3, E4\}$, with
$SituationSet = P\left(OASIS_{Characteristic}\right)$

($OASIS_{Characteristic}$ denotes the set of all influence factors in OASIS.)

We identified three basic relations that allow us to capture domain-knowledge in OASIS. The relations are depicted in Figure 3.15 and will be explained in more detail in the following:

1. **Imply**: $Xa \Rightarrow Y$
   The implication of characteristics has the semantics that if we have a situation containing Xa, then Y is implicitly included in the situation as well. Additionally, we can observe that the exposure value does not change if "Y" is (explicitly) included or not. An example of the characteristics implication is:
   $"driving" \Rightarrow "VehicleSpeed > 0km/h"$
   Given this implication, the following holds:
   $Exposure\left(SIT \cup \{"driving"\}\right)$
   $= Exposure\left(SIT \cup \{"driving", "VehicleSpeed > 0km/h"\}\right)$
   where SIT is a set of characteristics not containing $"driving"$ or $"VehicleSpeed > 0km/h"$ Please note that similar to the common definition of an implication, this relation is unidirectional! In $"a" \Rightarrow "b"$ it adds only information to "a"; "b" is not affected by this relation!

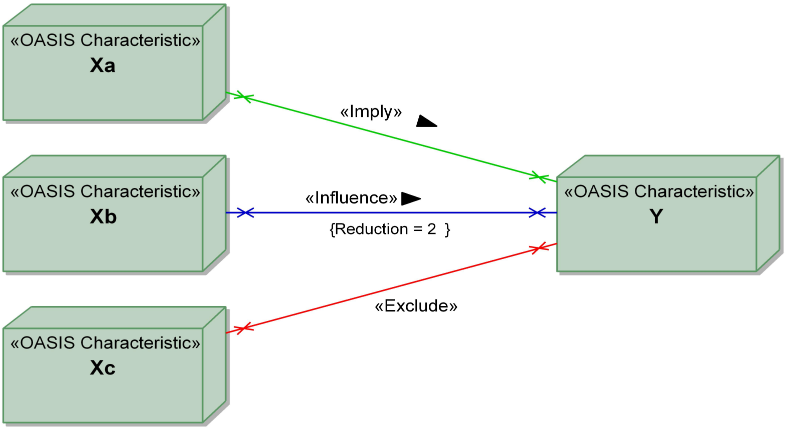


Figure 3.15: Knowledge capturing relations in OASIS

2. **Exclude**: $Xc \neg Y$
   Means that the concomitance of two characteristics does not make sense. If one were to assess the exposure of two characteristics with the exclude-relation, the result would always be E0 (meaning unlikely).
   An outside temperature of $50°\mathrm{C}$ and ice on the road, for example, is very unlikely. Thus, given:
   $"ice" \neg "50°\mathrm{C}"$
   the following always holds:
   $Exposure\,(SIT \cup \{"ice", "50°\mathrm{C}"\}) = E0$
   Please note that this relation is bidirectional, meaning the characteristics related to "exclude" are mutually exclusive!
3. **Influence**: $Xb \xrightarrow{red=x} Y$
   The influence relation just states that the probability of Y depends on the existence of Xb. It is, for example, quite probable that a car driving on a highway has a vehicle speed of >100km/h. A car driving in the city, however, does not usually reach a speed of >100km/h. How much the relation influences the probability is expressed with an "exposure reduction" parameter. One could, for example, define that, given a certain probability of driving up a mountain pass, this probability is reduced by two orders of magnitude if one includes the traffic situation "stop and go":
   $"mountainpass" \xrightarrow{red=2} "stopandgo"$

With the above-defined relations, we introduced the possibility of defining relations between concrete influence factors. While this makes sense in theory, in practice it is quite laborious to think about each combination of influence factors. In order to reduce this number, we did a pre-analysis on the concept or class level to decide in general whether it makes sense to analyze concomitances of influence factors of these classes.

If we take, for example the classes ***"Static Characteristic"*** and ***"Natural Condition"***, the question is whether we can define one or more relations between these two classes. Is it possible to define probability dependencies of ***"Natural Conditions"*** from ***"Static Characteristics"***? This is obviously not the case: Neither can a static characteristic, such as ***"roof luggage rack"***, change the probability of certain weather conditions, nor can the weather (significantly) influence the setup of a car. The classes ***"Action"*** and ***"Usage location"***, in contrast, have obvious dependencies: ***"Highway"*** and ***"Driving backwards"*** is obviously not likely. Thus, between our identified relevant classes of characteristics there are certain argumentative dependencies with respect to concomitance probability.

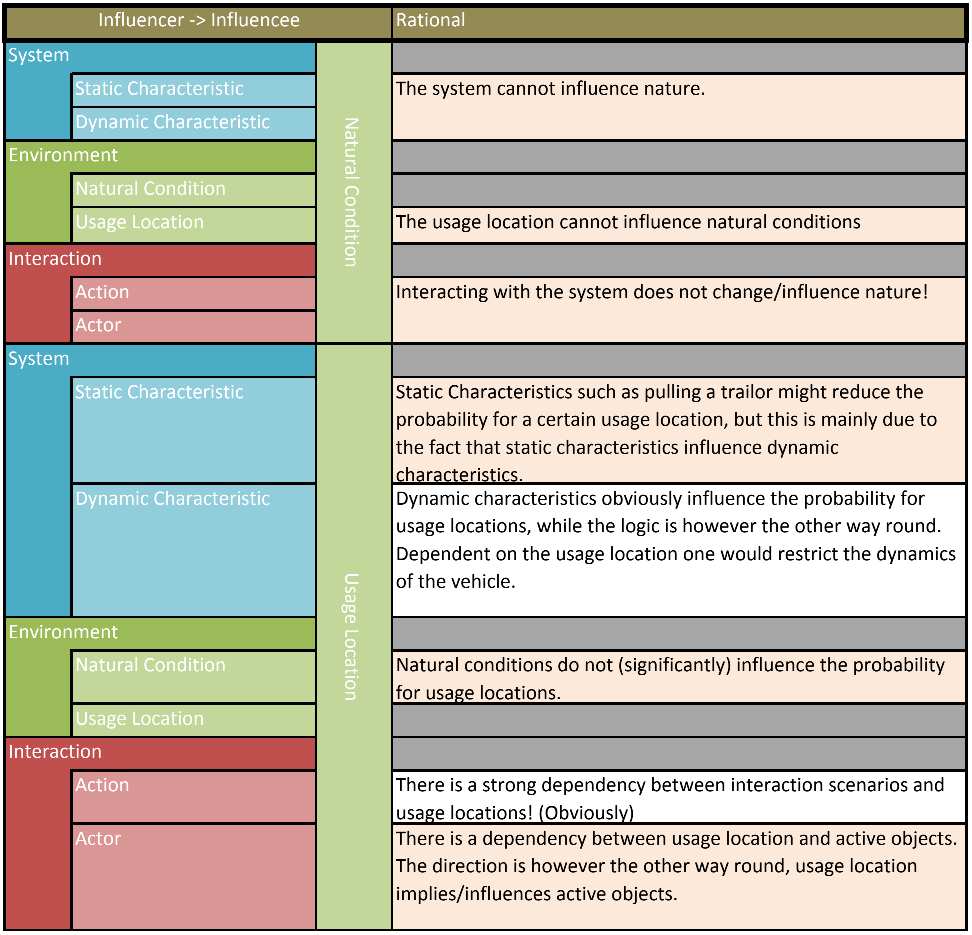
In order to grab the domain knowledge at the conceptual level (rather than at the influence factor level), we created an influence analysis matrix, which is depicted in Table 3.1.

The idea of the table is to systematically analyze the influence of each second-level class on all other second-level classes. At the far left of the figure are several blocks containing the general OASIS classes. In the middle, each (second-level) class appears again next to one block. Each entry in the left block is analyzed for potential influences on the characteristic in the middle. The rationale can then be found on the right. The light orange rationale cells are concomitances that are independent, meaning it does not make sense to define any relations between influence factors from these classes. An example of this would be our example of natural conditions and static vehicle characteristics. If the cell is white, however, then there is a dependency. The text in the cells gives a rationale for the color.

Table 3.1: Influence factor dependency analysis table

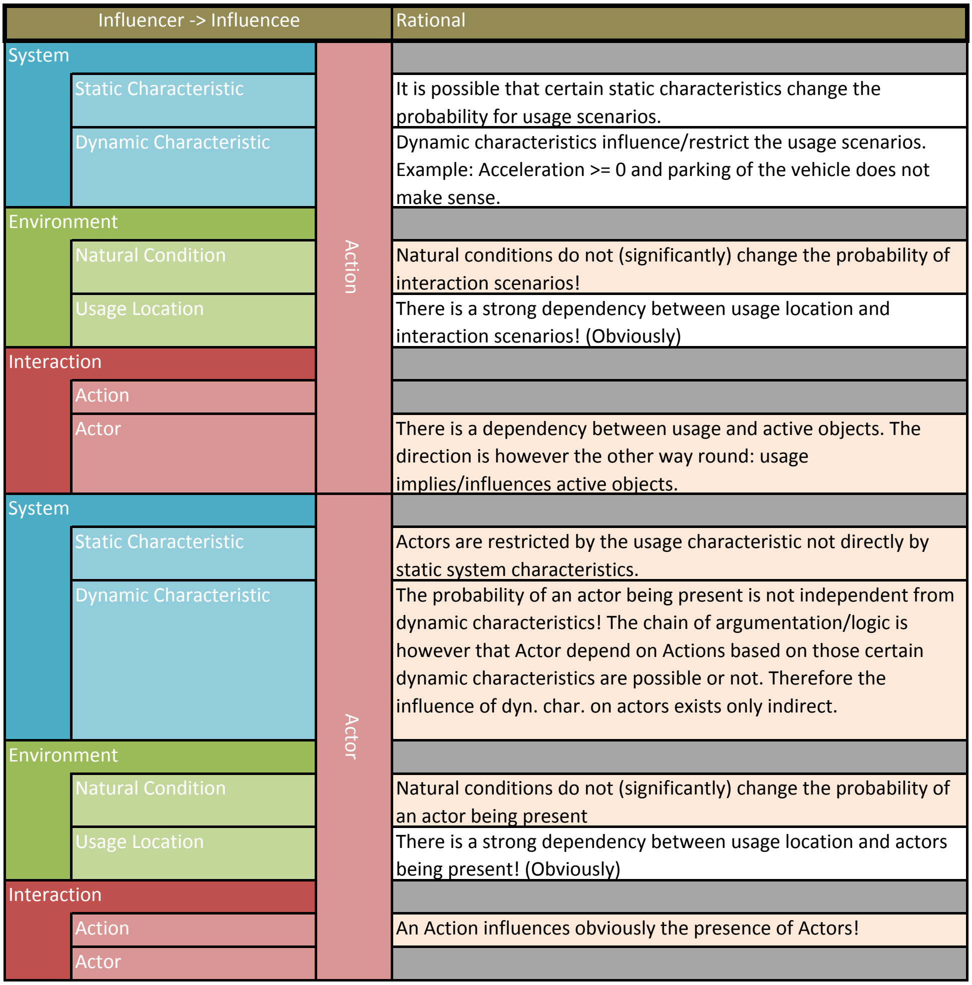
Influencer -> Influencee			Rational
System		Static Characteristic	
	Static Characteristic		
	Dynamic Characteristic		Static Characteristics is a pre-set system layout and cannot be influenced/changed. Therefore there is no influence from any other characteristic to Static Characteristics.
Environment			
	Natural Condition		
	Usage Location		
Interaction			
	Action		
	Actor		
System		Dynamic Characteristic	
	Static Characteristic		Static Ch. can influence the likelihood for certain dynamic characteristics. Pulling a trailer and driving more than 130 km/h is not very likely (against the law).
	Dynamic Characteristic		
Environment			
	Natural Condition		Natural conditions can influence vehicle dynamics! Example: Ice and slip (µ)
	Usage Location		The usage location obviously influences the probability of dynamic vehicle characteristics
Interaction			
	Action		Actions and dynamic characteristics are strongly dependent!
	Actor		Actors are mainly dependent on the usage location. The usage location again restricts the dynamic characteristics! Thus there is only an indirect influence.

Table 3.1: Influence factor dependency analysis table (continued)



Influencer -> Influencee			Rational
System		Natural Condition	
	Static Characteristic		The system cannot influence nature.
	Dynamic Characteristic		
Environment			
	Natural Condition		
	Usage Location		The usage location cannot influence natural conditions
Interaction			
	Action		Interacting with the system does not change/influence nature!
	Actor		
System		Usage Location	
	Static Characteristic		Static Characteristics such as pulling a trailor might reduce the probability for a certain usage location, but this is mainly due to the fact that static characteristics influence dynamic characteristics.
	Dynamic Characteristic		Dynamic characteristics obviously influence the probability for usage locations, while the logic is however the other way round. Dependent on the usage location one would restrict the dynamics of the vehicle.
Environment			
	Natural Condition		Natural conditions do not (significantly) influence the probability for usage locations.
	Usage Location		
Interaction			
	Action		There is a strong dependency between interaction scenarios and usage locations! (Obviously)
	Actor		There is a dependency between usage location and active objects. The direction is however the other way round, usage location implies/influences active objects.

Table 3.1: Influence factor dependency analysis table (continued)



Influencer -> Influencee			Rational
System		Action	
	Static Characteristic		It is possible that certain static characteristics change the probability for usage scenarios.
	Dynamic Characteristic		Dynamic characteristics influence/restrict the usage scenarios. Example: Acceleration >= 0 and parking of the vehicle does not make sense.
Environment			
	Natural Condition		Natural conditions do not (significantly) change the probability of interaction scenarios!
	Usage Location		There is a strong dependency between usage location and interaction scenarios! (Obviously)
Interaction			
	Action		
	Actor		There is a dependency between usage and active objects. The direction is however the other way round: usage implies/influences active objects.
System		Actor	
	Static Characteristic		Actors are restricted by the usage characteristic not directly by static system characteristics.
	Dynamic Characteristic		The probability of an actor being present is not independent from dynamic characteristics! The chain of argumentation/logic is however that Actor depend on Actions based on those certain dynamic characteristics are possible or not. Therefore the influence of dyn. char. on actors exists only indirect.
Environment			
	Natural Condition		Natural conditions do not (significantly) change the probability of an actor being present
	Usage Location		There is a strong dependency between usage location and actors being present! (Obviously)
Interaction			
	Action		An Action influences obviously the presence of Actors!
	Actor		

The analysis result discussed above can now be used to derive relations between the (second-level) classes of OASIS. This was done by analyzing each white cell on the right (in Table 3.1) and assigning possible relations. The result of this analysis is depicted in Figure 3.16. The kind of relation is defined by a short notation: an ordered set, or tuple. The first element refers to an ***"Influence"***, the second to an ***"Exclude"***, and the third to an ***"Imply"*** relationship. Hence " $(I, \_, I)$ " means that ***"Influence"*** and ***"Imply"*** relations are possible, but ***"Exclude"*** relations cannot be used in this case.

While all of these relations are captured in the ontology, inputting data into an ontology is time-consuming. It becomes evident that ontologies are not

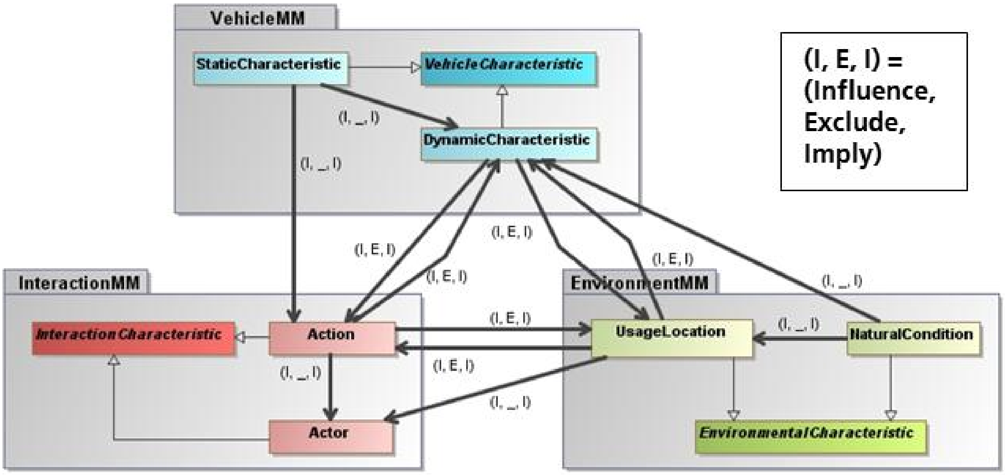


Figure 3.16: Influence factor class dependency graph

made for modelers (cf. first section of 3.3). Our claim is to keep an eye on practicability, which is why we created a modeling frontend for the definition of ontology models and especially the modeling of influence factor relations.

The domain knowledge modeler - a meta-model approach

In section 3.3.1, we discussed the difference between ontologies and meta-models. The main difference is that meta-models define a concrete grammar of how to build models while ontologies only define valid concepts (without construction information). For HRA applications, we want to use the situation information; thus we do not need construction instructions. However, if we want the domain expert to update the ontology and integrate knowledge, we have to provide an interface he/she can easily and efficiently work with. Therefore, we provide a modeling frontend for specifying the domain-knowledge-preserving relations introduced above!
For the realization of the modeling frontend we used an industry-strength UML modeling tool (MagicDraw) and customized the tool for our needs. The customization was done by wrapping the modeling-relevant parts in a meta-model. The necessary modeling elements are the structural elements of the ontology and the relations between instances of this ontology. The meta-model is shown in Figure 3.17:

It can be seen in the meta-model that we used the UML type "*Package*" to represent our ontology concepts, implemented as a specialization named "*OASISConcept*". This represents the notion of being a kind of container for similar HRA influence factors. Furthermore, it supports the nesting of packages, thus, the building of hierarchies of packages, which is, of course, helpful to represent our ontology.

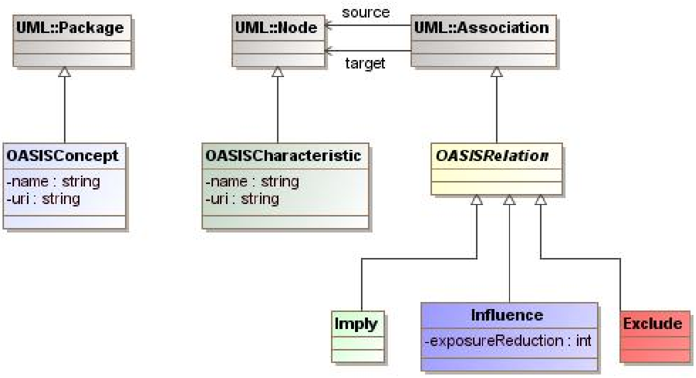


Figure 3.17: Meta-model (extension) for HRA domain knowledge modeling

The concrete instances, "*OASISCharacteristic*"s, were derived from the UML type "*Node*". The decision to use the "*Node*" type was mainly based on optical/graphical reasons because a "*Node*" is usually represented as a block, whereas a class is a rectangle with compartments (for attributes and methods). The visual simplicity and the analogy to "building blocks", which are our influence factors for HRA situations, was another factor contributing to our decision.

Figure 3.18 shows an example of the OASISConcept hierarchy and OASISCharacteristic. Maybe most important is the class "*OASISRelation*",

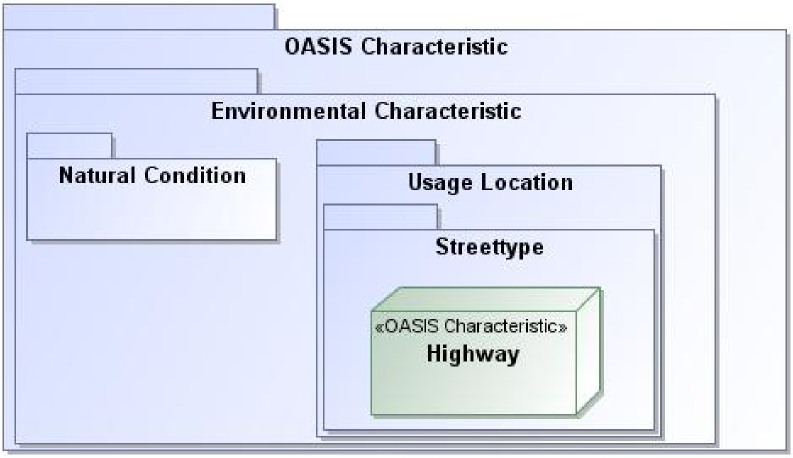


Figure 3.18: Example of OASISConcept and OASISCharacteristic modeling

which is a specialization of the UML type "Association". In our case, this is an abstract class, with the three specific refinements "*Imply*", "*Influence*", and "*Exclude*". The class "*Influence*" additionally has an attribute "*exposureReduction*" of the type "*int*". With this we can model an attributed association and use the exposureReduction attribute to define the influence strength introduced above.

With our meta-model it is possible to represent all (HRA-essential) information in an UML tool. To actually use the defined relations in a UML tool, it is necessary to import the ontology into the tool. Due to the "*Package*" type, this looks organized similarly to the ontology editor. Figure 3.19 shows a comparison of the hierarchy representation in the ontology modeler on the left and the UML tool on the right.

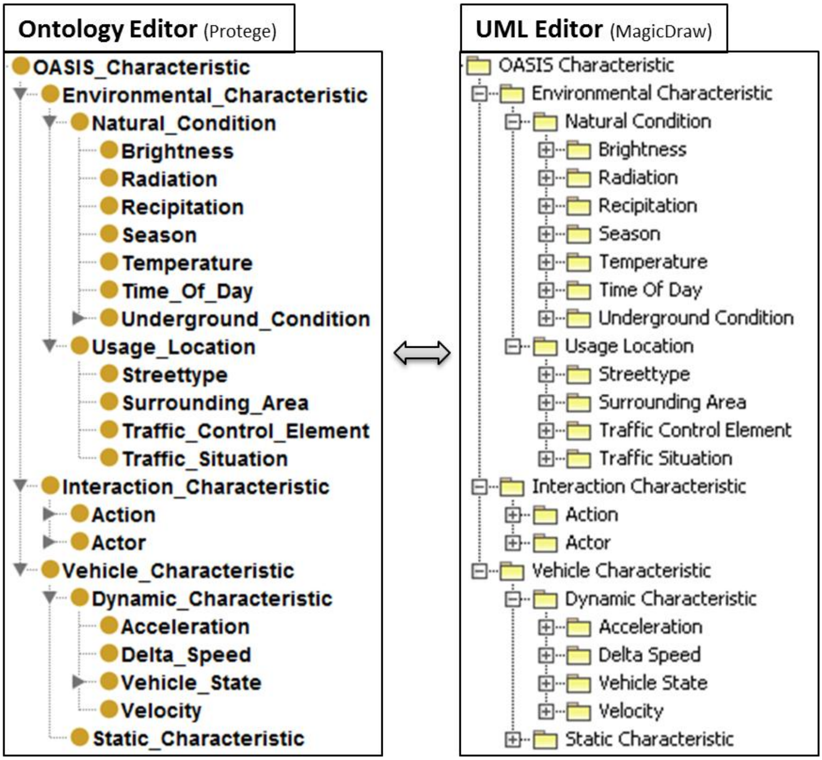


Figure 3.19: Comparison of hierarchy representations in an ontology tool and a UML tool

The usage of the meta-model for its purpose of modeling relations is shown in the subsequent figures.

Figure 3.20 shows the dependencies between influence factors of the classes ***"Street type"*** and ***"Surrounding Area"***. Thus, it is an influence factor relation within one second-level class (within Usage Location). Note that we layout the model so that ***"Street type"*** is in the middle and ***"Surrounding Area"*** is both above and under ***"Street type"***. We use the above represen-

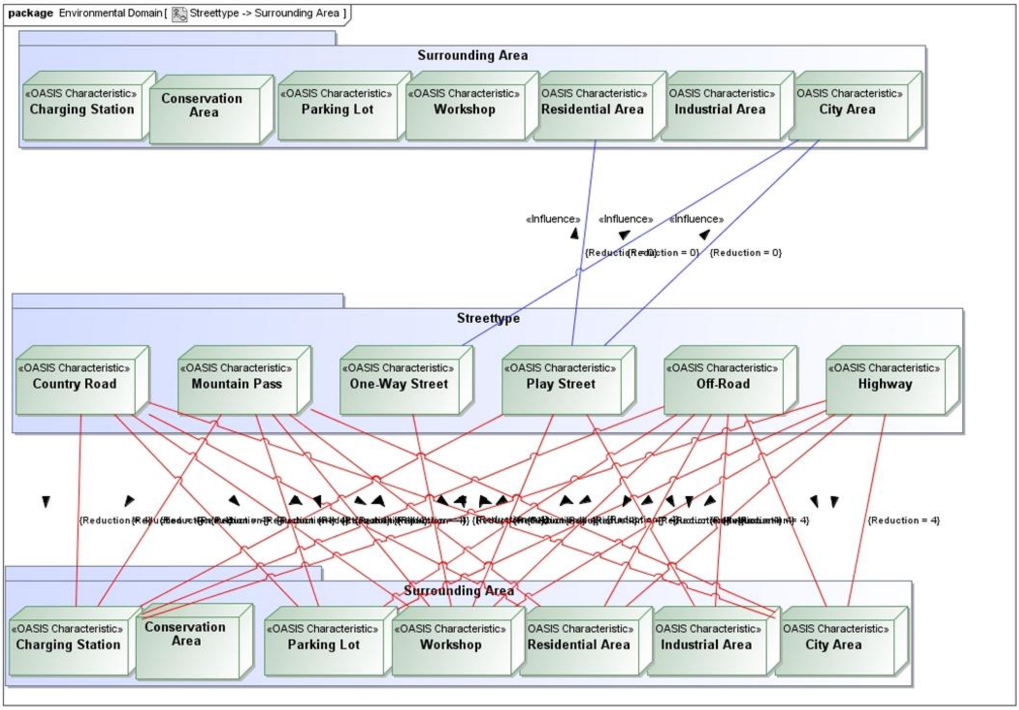

package Environmental Domain [Streettype -> Surrounding Area]
Surrounding Area
«OASIS Characteristic» Charging Station
Conservation Area
«OASIS Characteristic» Parking Lot
«OASIS Characteristic» Workshop
OASIS Characteristic» Residential Area
OASIS Characteristic» Industrial Area
OASIS Characteristic» City Area
«Influence»
«Influence»
«Influence»
{Reduction = 0}
{Reduction = 0}
Streettype
«OASIS Characteristic» Country Road
«OASIS Characteristic» Mountain Pass
«OASIS Characteristic» One-Way Street
«OASIS Characteristic» Play Street
«OASIS Characteristic» Off-Road
«OASIS Characteristic» Highway
{Reduction = 4}
Surrounding Area
«OASIS Characteristic» Charging Station
Conservation Area
«OASIS Characteristic» Parking Lot
«OASIS Characteristic» Workshop
OASIS Characteristic» Residential Area
OASIS Characteristic» Industrial Area
OASIS Characteristic» City Area

Figure 3.20: Dependency of influence factors "Street type" and "Surrounding Area"

tation to model the "***Influence***" relation and the lower model representation to model the ***"Exclude"*** relations.

This is, however, only a convenient way to keep things clearly separated. It is neither enforced nor restricted by the tool.

Figure 3.21 represents an example of the ***"Imply"*** relation by showing the dependency modeling of ***"Operator"*** and ***"Street type"***.

Due to the simplicity of this case, we used a simple layout where the influenced class ***"Operator"*** appears only above the influencing class ***"Street type"***. The green association lines (the four lines going to the driver) are ***"Imply"*** relations, the others (the red ones) are ***"Exclude"*** relations.

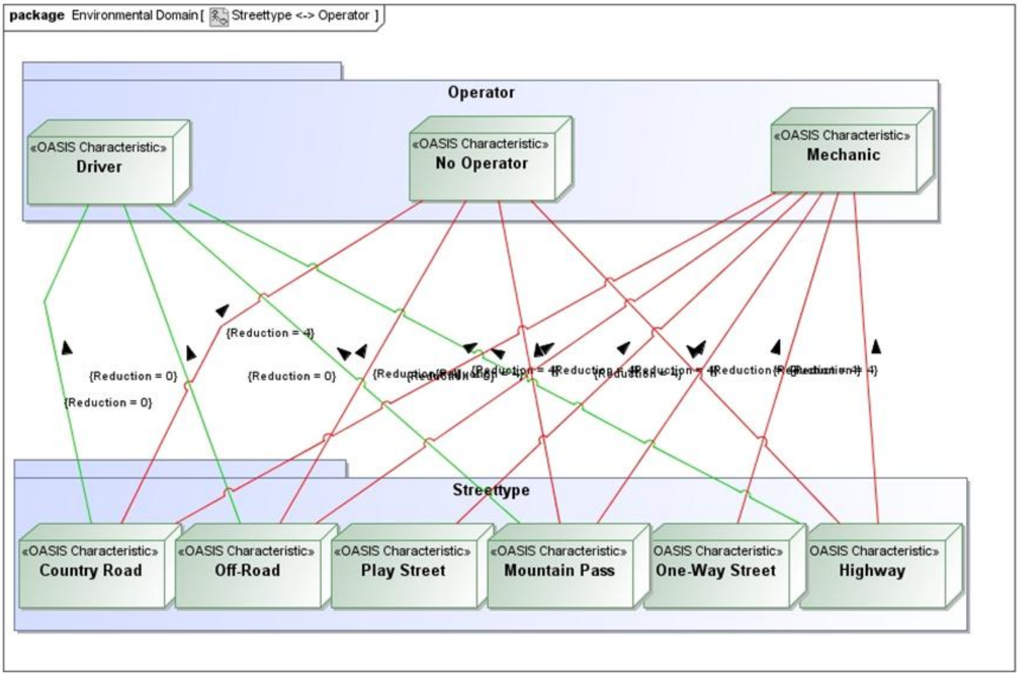

package Environmental Domain [Streettype <-> Operator]
Operator
«OASIS Characteristic» Driver
«OASIS Characteristic» No Operator
«OASIS Characteristic» Mechanic
{Reduction = 4}
{Reduction = 0}
{Reduction = 0}
{Reduction = 0}
Streettype
«OASIS Characteristic» Country Road
«OASIS Characteristic» Off-Road
«OASIS Characteristic» Play Street
«OASIS Characteristic» Mountain Pass
OASIS Characteristic» One-Way Street
OASIS Characteristic» Highway

Figure 3.21: Dependency of influence factors "Street type" and "Operator"

3.3.4 Situation definition process

In the previous sections of this chapter, we discussed how HRA situations can be formally represented and how their building blocks, the influence factors, are classified and structured in an ontology. Additionally, we have shown how one can capture domain knowledge as relations between influence factors.

The purpose of modeling situations, however, is to use them in a concrete HRA application. Thus, the remaining question is how a safety engineer can use the ontology in order to define HRA-relevant situations.

The requirement 7.4.2.1.1 in ISO 26262-3:2011 is: *"The operational situations and operating modes in which an item's malfunctioning behavior will result in a hazardous event shall be described, both for cases when the vehicle is correctly used and when it is incorrectly used in a foreseeable way."*

It is stated that one should create an exhaustive list of HRA situations (including operating modes). Formally speaking, the demand is that all sufficient situations should be created. A sufficient situation means in this context that the situation in concomitance with *"an item's malfunctioning behavior will result in a hazardous event"*. The additional requirement 7.4.4.2 states:

"It shall be ensured that the chosen level of detail of the list of operational situations does not lead to an inappropriate lowering of the ASIL of the corresponding safety goals".

This requirement adds the extension that the list should only contain a level of detail that is required to distinguish different hazardous situations. It adds the aspect of being necessary to the set of influence factors of a situation definition.

The demand is to define for each hazard a complete set of HRA situations, where each situation contains sufficient and (only) necessary influence factors!

The safety expert has to make the decision about the sufficiency and necessity of influence factors for each situation. SAHARA can help the safety engineer by guiding him in this definition process. This section will discuss this guidance by introducing a situation definition process.

We structure this section into three parts. First, we will transfer the so-called "Meta-Model" approach (for precise communication) by Grinder and Bandler [Gri76] to the elicitation of situations. This unveils pitfalls and basic principles for the definition of situations (for HRA). Afterwards we will apply this to HRA situation models and propose a specific process for the definition of HRA situations. We will finalize this section by providing a running example showing the situation definition with OASIS.

Precise mental models - Fundamentals

In GOBI, we presented a conceptual model describing an interaction of participants of an HRA explicitly. A safety engineer has a similar model in mind when he performs the task of an HRA. The purpose of this section is to understand and describe the way a safety engineer thinks or should think to come up with a good concrete model of an interaction sequence of the FVMs. The main process of handling information mentally, it to cluster and chunk elements [Mil56]. For us this means that a picture (the model) should be built, starting from coarse-grained, general information pieces and ending with more refined ones. In the case of situation definitions, we need to set up the big picture first, before adding details.

One experience we made during our industrial case studies is that some experts "jump" to conclusions and omit the definition of concrete situations. It seems to them that the information is not "necessary". The reasons for that are the two mental processes "deletion" and "generalization". These phenomena are well-known in psychology [Ban75], as well as in requirements engineering [Rup04]:

- *"Deletion"* is in most cases a mental presupposition. This means that the expert omits details because they are implicitly clear or obvious to him. For the HRA situation definition "deletion" means that an expert (intentionally) omits relevant details. The "deletion"-case we found most often is to omit concrete details of a situation and state only physical characteristics. Physics is good because it is precise and concrete, but by focusing on and thinking in the physical failure world, one is often not open for other influences and therefore for new situations (with another physical effect). The danger of "deletion" is that potentially *necessary* influence factors are not considered.

  One example of deletion is that the car is "locked". A situation from an industrial case study in the context of the parking brake revealed the situation where the car is "locked" and the car is parked on a hill. The scenario was that the parking brake fails and the car starts to move. This is a valid case. But the implicit assumption (the occurrence of deletion) was that the expert implied that there is no one in the car and that no one can quickly jump into the car in order to control the situation. The implication was that locking a car means that there is no one inside the car or close to the car anymore. Not stating this explicitly is a case of deletion. Deletion often happens due to inherent domain knowledge. If we figure out deletion cases, we can use the OASIS relation "imply" to explicitly capture this in the ontology. Once this information is captured, "deletion" occurrences (in this particular case) are handled by the ontology.
- *"Generalization"* refers to incorrect abstraction. While abstraction helps to handle complexity, incorrect abstraction means that we make a mistake by not considering enough details. For the HRA situation definition, "generalization" means that an expert (unintentionally) omits relevant details. The "generalization" case we found most often is to abstract

from details and therefore make the set of situations incomplete or the affected situations not *sufficient*.

An example of the generalization issue from an industrial case study is related to the (active) power steering system. The discussion was about omission failures of the system. I.e., the vehicle does not apply a steering angle upon request. Many concrete situations were discussed, such as "driving/entering a curve on a highway", "cornering in the city", or "driving on a winding road on a mountain pass". In the analysis, all this was reduced to the single influence factor "curved road". While the primary common aspect is indeed a curved road, the incorrect abstraction was to omit the fact that the car needs to be driving! If it is parked on a curved road, a steering omission is not critical. Thus, in this case the intention to focus on the essential aspect by generalizing or abstracting from concrete situations made the resulting situation not sufficient. One could argue that this is also a case of deletion, in which the implication of "curved road" to "is moving" is deleted. However, this is wrong because the implication is not generally true. What one might have in mind is an implication (and therefore a "deletion") of "all considered situations imply moving". The logic is that:

"all considered situations" $\overset{imply}{\Longrightarrow}$ "curved road" and

"all considered situations" $\overset{imply}{\Longrightarrow}$ "moving"

The *wrong* assumption is that based on this knowledge, the conclusion is drawn that:
"curved road" $\overset{imply}{\Longrightarrow}$ "moving"

The basic principles we should integrate into our situation definition process are therefore:

- Start with the big picture by stating coarse-grained information
- Be specific! State even "common" knowledge/sense.
- Be complete! Do not take "short-cuts" through abstraction.

These principles were the fundamental things we considered when we developed the HRA situation definition process presented in the next section.

The HRA situation definition process

Based on the principles introduced above, we defined our situation creation process as shown in Figure 3.22. We will explain each step and give some ideas and rationales for the proposed process as well as lessons learned during the development.

Step 1:
As stated above, the assumption is that we have identified a concrete item failure or more generally a hazard.

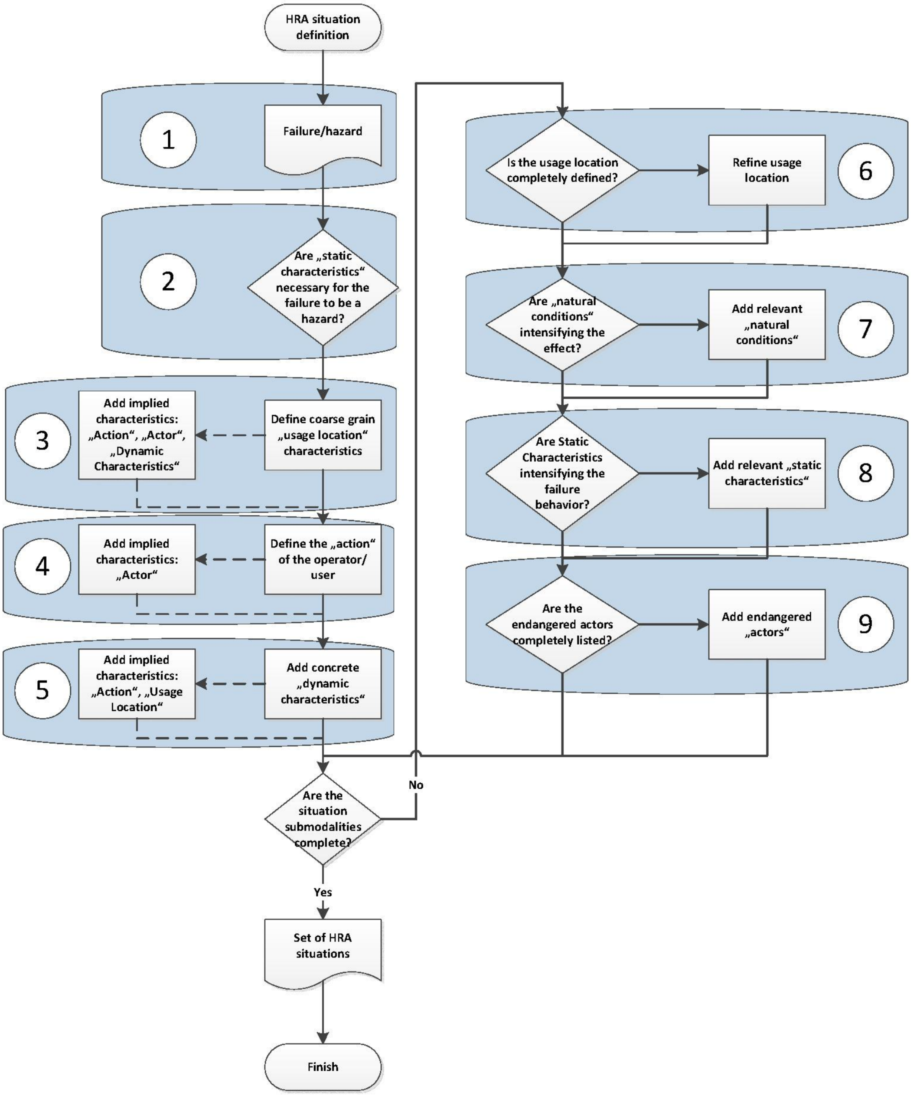

HRA situation definition
1
Failure/hazard
2
Are „static characteristics“ necessary for the failure to be a hazard?
3
Add implied characteristics: „Action“, „Actor“, „Dynamic Characteristics“
Define coarse grain „usage location“ characteristics
4
Add implied characteristics: „Actor“
Define the „action“ of the operator/ user
5
Add implied characteristics: „Action“, „Usage Location“
Add concrete „dynamic characteristics“
6
Is the usage location completely defined?
Refine usage location
7
Are „natural conditions“ intensifying the effect?
Add relevant „natural conditions“
8
Are Static Characteristics intensifying the failure behavior?
Add relevant „static characteristics“
9
Are the endangered actors completely listed?
Add endangered „actors“
No
Are the situation submodalities complete?
Yes
Set of HRA situations
Finish

Figure 3.22: HRA situation definition process

Step 2:
Sometimes a failure needs some vehicle configuration/setup to transition into a hazard or it simply does not make sense to think about the functional failure because the function could not work without a certain vehicle setup. The driver assistance function "trailor esc", for example, needs a "trailer" as

a static characteristic in order to make sense (at least for all non-commission failures). The static characteristics describe a general system structure in more detail. This aspect is often used to describe variants of systems or products. Thus, if one is interested in an HRA product line, this would be the location for defining variation points. Note that HRA product lines are not in the scope of this thesis.

Step 3:
Once the concrete (static) model of our vehicle is defined and one has the hazard in mind, it is important to get a clear picture of the surroundings. In our case, the influence factors of relevance are the "Usage Location" influence factors. What is really essential is the coarse-grained "big picture"; thus, one should be careful not to over-specify the usage location at this point of time. We propose this due to the "necessity" requirement stated above. Our experience has shown that a good "subclass" to start with for coarsely defining the usage location is the subclass "Street Type": The number of street types is limited and everyone has a very clear picture of the various types of streets in mind. Thus, it constitutes a good "big picture".

Step 3 (left):
As already identified in the previous section, the "Usage Location" might have implications on "Action", "Actor", and "Dynamic Characteristic" influence factors. It is good practice to always maximize the influence factors because this prevents "deletion" problems and does not affect exposure (due to the definition of the "imply" relation). The expansion of the "implied" influence factors is shown on the dashed path to the left of the process block.

Step 4:
Once the big picture is defined (with "Usage Location" influence factors), we propose making the connection between the hazard and the big picture more explicit. This is done by stating the current maneuver in which the user or operator is using the vehicle. The rationale for this decision is that if one mentally pictures the hazard of the vehicle in the usage location, one usually already has certain maneuvers in mind.

In a previous version of our work, we switched the process steps: define the action/maneuver first and then integrated the usage location. However, it turned out that it is easier to imagine a usage location first and define a maneuver afterwards. If one is "forced" to state a maneuver first, he mentally pictures a usage location and derives the respective maneuver from that. Due to this reason we switched these steps to the order introduced above.

Step 5:
Up to now we have the system misbehavior (1), with essential system characteristics (2), in a (coarse-grained) environmental setting (3) along with the intended action (the maneuver) (4). The next step is to go back to the vehicle and add some dynamics information. In this step, concrete physical properties of the vehicle are described. These dynamic aspects constitute a rationale for the harm and are often used as an argument for the severity

of the harm. This is not too surprising because dynamics are about energy and energy (mostly kinetic energy) is our "potential source of harm".

Now the situations are almost complete (except for some details). The (potentially) missing details do not substantially change the situation, but perfect the situation with respect to the trade-off between exposure and severity (and controllability) (as discussed in section 4.2 GOBI). We term the situations defined up to this step *"basic situations"*.

Step $(6 \rightarrow 7 \rightarrow 8 \rightarrow 9)$*:*
The polishing process starts with the decision of the expert about whether the environment, i.e., the coarse-grained surrounding definition is hazardous "enough" to be harmful. If this is not the case, he should add necessary details of the surrounding, but only if these are definitely necessary for the harm. The usual approach is to first detail the usage location (6) and afterwards add natural conditions (7).

The next steps are to reduce the controllability or increase the severity. One can decrease controllability by specifying additional static characteristics of the vehicle (8). Severity is often related to how people are harmed. We can add this by adding additional "endangered" actors (9). Especially the passive, but endangered actors are often forgotten (due to the "deletion" phenomenon). In order to make the situation consistent (and complete), one should check if all relevant actors are included in the situation.

After step (9) it is checked again whether the situation is complete wrt. its submodalities. If this is not the case, the polishing process $(6 \rightarrow 7 \rightarrow 8 \rightarrow 9)$ is iterated another time.

Once we leave the decision with a "yes", at the bottom, we have a complete list of sufficient and necessary situations for the hazards under investigation.

In the following section, we present an example application of this process. This should make it more concrete and tangible.

Evaluation and example: HRA situation definition

The example is taken from an *active power steering* case study. The basic functionality is that one can drive the car "by wire". This means that the steering requests are purely logical and the actual steering is done by controlling an actuator (via control software).

Step 1:
The failures we consider in our running example are shown in Figure 3.23.

Step 2:
In the example we do not have static characteristics. One might extend the features of the active power steering by disabling the active steering if a trailer is pulled (as a hypothetical case). Then we would add "pulling a trailer" as a static characteristic in this step. In order to keep the example

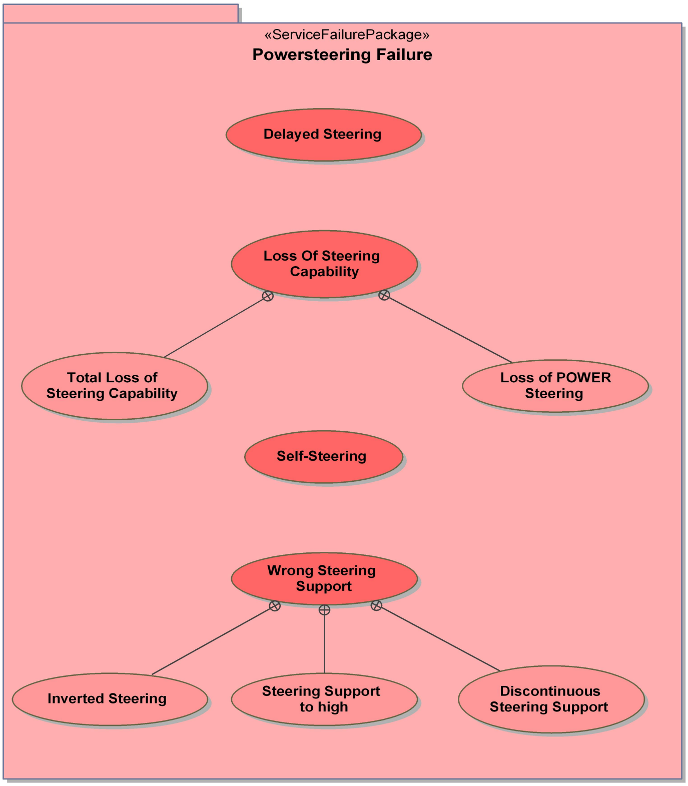


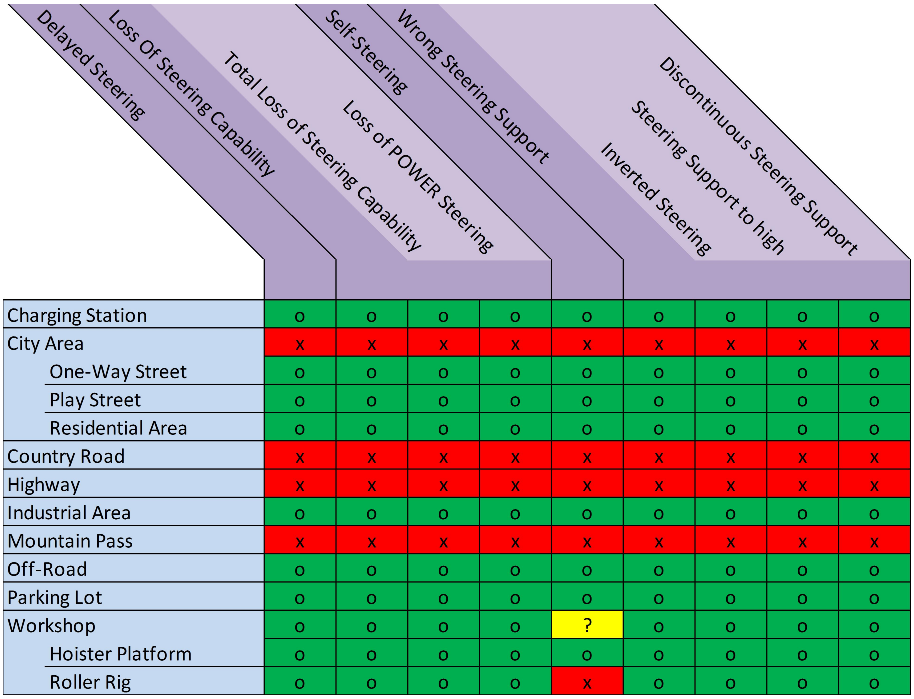
Figure 3.23: Example: Active Power Steering Failures

understandable, we omit this and do not add any static characteristics in this step.

Step 3:
In step 3, the coarse-grained usage location should be added. Figure 3.24 shows the considered or selected street types relevant in this example.

On top of the figure, the failures are shown. On the left concrete street types. The matrix is filled with "x"s and "o"s. x' on red background define unsafe combinations, i.e., situation setups we need to further investigate. Additionally a question mark on a yellow background is shown. This indicates that the workshop "location" itself is neither safe nor unsafe per se. In order to decide its relevance, we have to use a more refined view on the "workshop". We have to select one of the refined influence factors. In this case, the self-steering failure on a roller rig is considered to be dangerous, whereas being lifted on a hoisting platform is not critical.

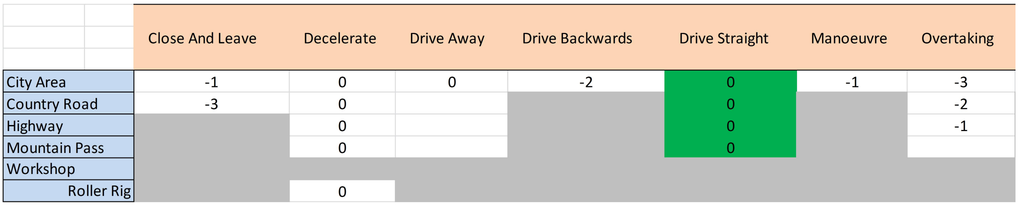
Step 3 (left):
In the running example, there are implications on "Action" influence fac-



	Delayed Steering	Loss Of Steering Capability	Total Loss of Steering Capability	Loss of POWER Steering	Self-Steering	Wrong Steering Support	Inverted Steering	Steering Support to high	Discontinuous Steering Support
Charging Station	o	o	o	o	o	o	o	o	o
City Area	x	x	x	x	x	x	x	x	x
One-Way Street	o	o	o	o	o	o	o	o	o
Play Street	o	o	o	o	o	o	o	o	o
Residential Area	o	o	o	o	o	o	o	o	o
Country Road	x	x	x	x	x	x	x	x	x
Highway	x	x	x	x	x	x	x	x	x
Industrial Area	o	o	o	o	o	o	o	o	o
Mountain Pass	x	x	x	x	x	x	x	x	x
Off-Road	o	o	o	o	o	o	o	o	o
Parking Lot	o	o	o	o	o	o	o	o	o
Workshop	o	o	o	o	?	o	o	o	o
Hoister Platform	o	o	o	o	o	o	o	o	o
Roller Rig	o	o	o	o	x	o	o	o	o

Figure 3.24: Example: Usage-Location selection

tors, on "Actors", and on "Dynamic Characteristic"s influence factors. Implications of influence factors can be domain knowledge and captured in the ontology. Figure 3.25 shows the domain-knowledge we reused from the ontology. The pre-selection is done completely automatically. Furthermore, note that due to spatial restrictions, not all maneuvers are shown in this figure. The overall number of maneuvers in our ontology is 15.

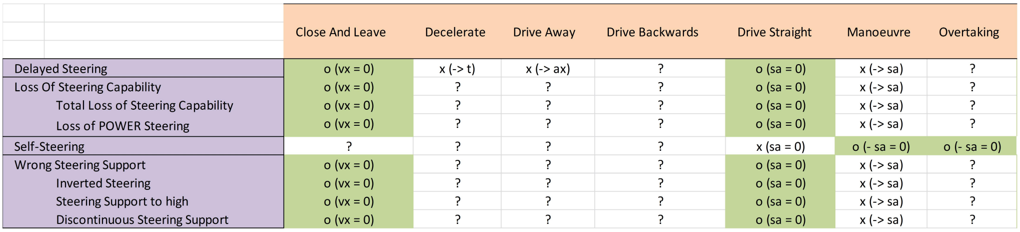


	Close And Leave	Decelerate	Drive Away	Drive Backwards	Drive Straight	Manoeuvre	Overtaking
City Area	-1	0	0	-2	0	-1	-3
Country Road	-3	0			0		-2
Highway		0			0		-1
Mountain Pass		0			0		
Workshop							
Roller Rig		0					

Figure 3.25: Example: "Action" relations of the selected Usage Locations

The gray cells are combinations of usage locations and maneuvers that do not make sense or are unlikely to occur. The white cells are either blank (then we do not have any further domain knowledge about this concomitance), or filled with a non-positive value. Those values are the values from the exposure reduction introduced above (of the OASIS "influence" relation). Driving backwards in the city, for example, is considered to be two orders of magnitude less likely than all city area situations together (meaning without restrictions). The default marker is shown by the green background in the column "Driving Straight". The default marker is similar (but not the same) as the "imply" relation. It means if nothing else is stated, we should assume this. Thus, if we do not state a concrete maneuver, one usually (implicitly) think of "Driving Straight" for the "City Area" for example. This allowed us to fully automatically reduce the possible number of combinations by 40% (from 138 to 85).

This pre-selection of possible maneuvers based on OASIS information does, however, not include information and a concomitance analysis with the hazard or the failure. For our running example this means that we are still in step (3): identify relevant maneuvers. For the consideration of hazards in the maneuver selection process, we can reuse GOBI-"knowledge". GOBI helps us because as part of the failure analysis in GOBI we identified the physical implications of failures, meaning the possibly affected physical values. This can be used to automatically restrict the number of relevant actions. The automated result is shown in Figure 3.26.Using OASIS (and GOBI) we could reduce the number of possible combinations (fully automatic) from 138 to 106 relevant combinations.

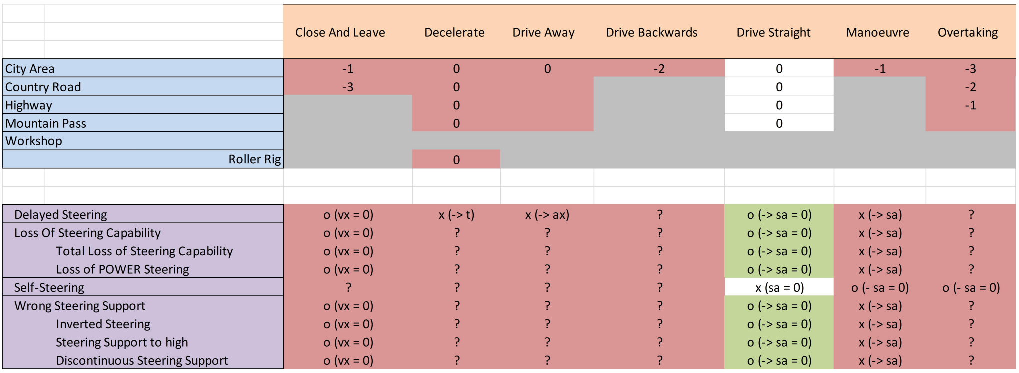


	Close And Leave	Decelerate	Drive Away	Drive Backwards	Drive Straight	Manoeuvre	Overtaking
Delayed Steering	o (vx = 0)	x (-> t)	x (-> ax)	?	o (sa = 0)	x (-> sa)	?
Loss Of Steering Capability	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?
Total Loss of Steering Capability	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?
Loss of POWER Steering	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?
Self-Steering	?	?	?	?	x (sa = 0)	o (- sa = 0)	o (- sa = 0)
Wrong Steering Support	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?
Inverted Steering	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?
Steering Support to high	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?
Discontinuous Steering Support	o (vx = 0)	?	?	?	o (sa = 0)	x (-> sa)	?

Figure 3.26: Example: "Action"-preselection due to GOBI's physics annotation

The cells with green background are "safe" combinations; the physical rationale is listed in brackets. The "Delayed Steering" hazard, for example, is not relevant in a "Drive Straight" maneuver because there is no steering request. The cells with the question mark denote combinations where we cannot derive any information from the physics annotation. The cells marked with an "x" are relevant and should be considered. The rationale for this table is shown in brackets behind the "x".

The next partial step is to integrate the relevant combinations extracted above into one set of possible/relevant combinations. The automatic calculation of relevant combinations of hazard, action, and street type, restricted the set of possible triple concomitances to 52 possibly relevant (out of 756 possible combinations). This is essentially the final result of the action analysis/description (Step 3). Figure 3.27 shows the result of the action selection process step.



	Close And Leave	Decelerate	Drive Away	Drive Backwards	Drive Straight	Manoeuvre	Overtaking
City Area	-1	0	0	-2	0	-1	-3
Country Road	-3	0			0		-2
Highway		0			0		-1
Mountain Pass		0			0		
Workshop							
Roller Rig		0					
Delayed Steering	o (vx = 0)	x (-> t)	x (-> ax)	?	o (-> sa = 0)	x (-> sa)	?
Loss Of Steering Capability	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?
Total Loss of Steering Capability	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?
Loss of POWER Steering	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?
Self-Steering	?	?	?	?	x (sa = 0)	o (- sa = 0)	o (- sa = 0)
Wrong Steering Support	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?
Inverted Steering	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?
Steering Support to high	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?
Discontinuous Steering Support	o (vx = 0)	?	?	?	o (-> sa = 0)	x (-> sa)	?

Figure 3.27: Example: Selection of relevant "Actions"

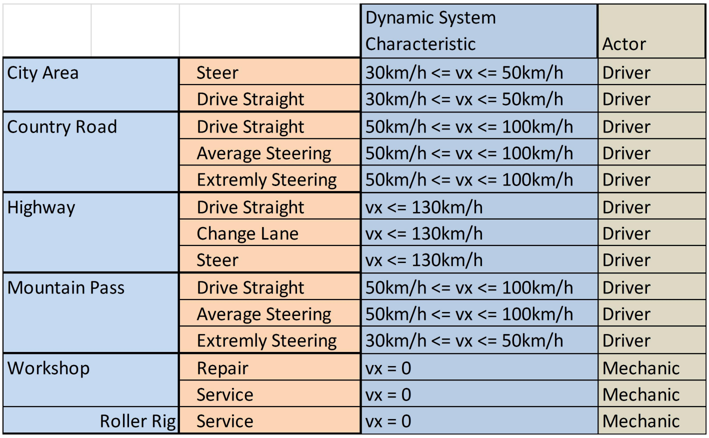
In the figure, only the background color changed from white to red. Red means concomitance not relevant (provably). In this fraction of maneuvers, or actions the only four relevant concomitances are represented by the combination of the one white cell in the lower part (the failure part) and the four white cells in the upper part (the usage location part). All other maneuvers can be proven to be irrelevant.

Step 5:
The next step is to add some vehicle dynamics. In the process presentation above, we referred to energies to be considered. In our running example, the energy mostly driving the harm is the vehicle's (longitudinal) velocity. The annotation of the velocity is shown in Figure 3.28.

The figure additionally shows the above-mentioned "Actor", which was already implied by step 4.

With this we have completed the situation definition. We could enter the polishing process. But for that we would have to state more assumptions, and we would need to know more details about the vehicle and its physics.

The process and especially the integration of OASIS and GOBI knowledge should have become clear in the steps presented above. As stated above, the further refinement is "tuning".



			Dynamic System Characteristic	Actor
City Area		Steer	30km/h <= vx <= 50km/h	Driver
		Drive Straight	30km/h <= vx <= 50km/h	Driver
Country Road		Drive Straight	50km/h <= vx <= 100km/h	Driver
		Average Steering	50km/h <= vx <= 100km/h	Driver
		Extremly Steering	50km/h <= vx <= 100km/h	Driver
Highway		Drive Straight	vx <= 130km/h	Driver
		Change Lane	vx <= 130km/h	Driver
		Steer	vx <= 130km/h	Driver
Mountain Pass		Drive Straight	50km/h <= vx <= 100km/h	Driver
		Average Steering	50km/h <= vx <= 100km/h	Driver
		Extremly Steering	30km/h <= vx <= 50km/h	Driver
Workshop		Repair	vx = 0	Mechanic
		Service	vx = 0	Mechanic
	Roller Rig	Service	vx = 0	Mechanic

Figure 3.28: Example: HRA situations of steering example

3.4 HEAT - Handy Exposure Assessment Technique

In the previous section, the OASIS section, we presented our formalization approach of essential influence factors and their dependencies for defining HRA situations. The contribution of the previous section was a formalization approach for operational situations. In this section, we use this formalization approach and extend it to capture the assessment of operational situations, the exposure parameter, as well. Figure 3.29 shows the integration of HEAT with the overall thesis on the one hand and the general structure of this section on the other hand.

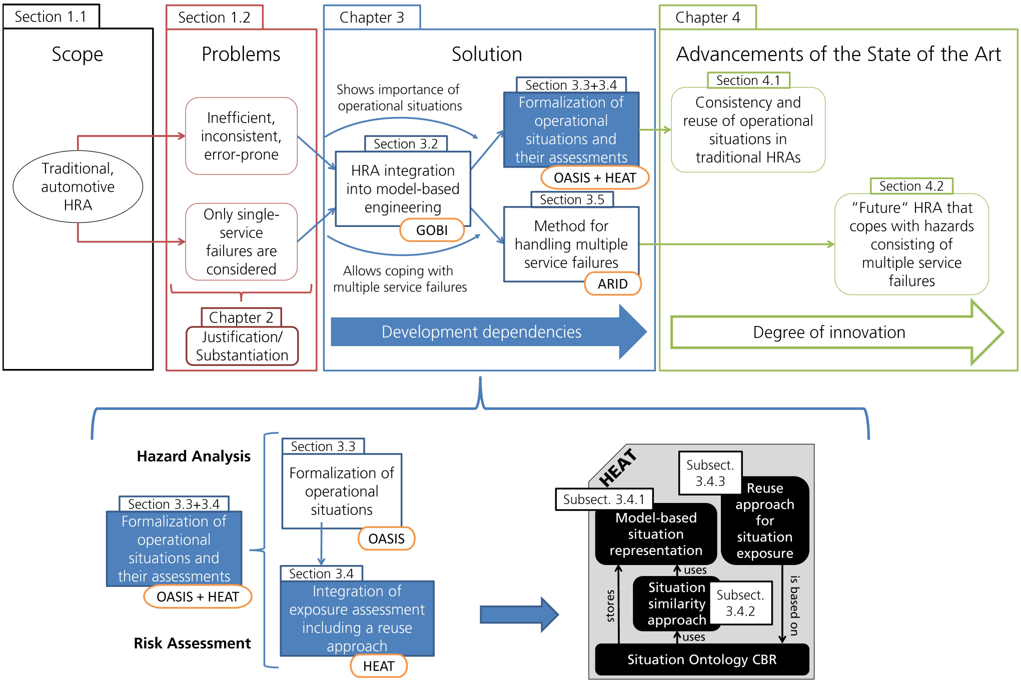

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method for handling multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation
Hazard Analysis
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Risk Assessment
Section 3.3
Formalization of operational situations
OASIS
Section 3.4
Integration of exposure assessment including a reuse approach
HEAT
HEAT
Subsect. 3.4.1
Model-based situation representation
Subsect. 3.4.3
Reuse approach for situation exposure
uses
Situation similarity approach
Subsect. 3.4.2
uses
stores
is based on
Situation Ontology CBR

Figure 3.29: Integration overview of HEAT

Besides the formalization of the environment, we identified as part of section "2.2 Core Challenges and Goals" another challenge:

✧ Structuring of hazard analysis

  ✧ Correctness of situation analysis

    ✓ Formalization of operational situations ↪ 3.3

    ➠ Efficient reuse of operational situations

Additionally, as part of the work related to "correctness of situation analysis" (cf. section 2.4.1), we derived the following findings, which we will transfer to the application of efficient reuse of situations.

1. Based on Ruben Prieto-Diaz' work [PD01] about reuse, we discussed the two extremes of either building everything we would like to reuse upfront, or building everything upon demand, starting with an empty set of reusable items. The finding of this discussion was that both extremes are impractical for a reuse approach of HRA situations, and we therefore proposed a reuse approach that starts off with an initially filled set of essential/basic HRA situations and lets the reuse database grow over time.
2. One technological finding we discussed was that case-based reasoning is highly appropriate for our reuse scenario. The basic idea is to use old cases and try to derive knowledge (lazily) if a new case comes up [Lea94]. In our domain, a case is an assessed HRA situation, and the most important reuse factor is the assessment value.
3. Recio-Garcia presented in [RgDaGc^{+}06] the connection between ontologies and case-based reasoning. Due to the fact that our influence factors are defined in an ontology (cf. OASIS), this finding fits very well into our solution approach, and we will reuse his ideas in order to technically integrate our reuse approach and situation modeling.

As already stated as part of item 2, the essential part in reusing a situation is not the HRA situation description but, more importantly, the exposure value. "Exposure assessment" was another challenge we identified in section "2.2 Core Challenges and Goals":

- ✧ Structuring of risk assessments
  - ✧ Hazardous event classification
    - ➠ Exposure assessment

While there is no directly related work for exposure assessments, we discussed in section "2.4.1 Correctness of situation analysis" the possibilities for coming up with an exposure value. The results of this discussion were:

1. It is not (practicably) possible to automatically calculate situation exposure values. A safety (and domain) expert always needs to decide on the exposure values personally. This has technological as well as legal reasons.
2. We can support the decision process of an "exposure assessment" by providing the safety engineer with similar cases he can base his decision on (or just backup his decision).

Thus, in this section we present our solution for reusing (essentially the exposure value of) HRA situations. The solution approach is subdivided into the following structure, which directly leads to the subsequent structure of this section:

1. First of all we need a representation of an (assessed) HRA situation. We therefore introduce an HRA situation model extension of our OASIS influence factor ontology. This is the subject of section "3.4.1 HRA Situation

modeling". Note that in that section we will only discuss the elements of an assessed HRA situation, such as the assessment result (exposure value); the process for coming up with an exposure value will be defined in a later section.

2. Once we have a formal representation of HRA situations, we will discuss the first step towards reusing a situation: the retrieval of already existing situations. We will therefore present a reuse approach that combines ontologies and case-based reasoning by defining situation similarity. (Case similarity is one of the most important aspects in case-based reasoning.) We will discuss this in section "3.4.2 HRA Situation similarity".
3. Having defined HRA situation similarity, we can apply the case-based reasoning technique for our fundamental goal of reusing assessed HRA situations. The reuse process essentially supports the safety/domain expert in the situation assessment process. How this is actually done is the subject of section "3.4.3 Exposure Assessment Support".

After reading this section, the reader will know how HRA situations are represented in our ontology-based approach and, even more importantly, how he can use this representation to efficiently reuse HRA situations and their exposure assessment result.

3.4.1 HRA Situation modeling

As described in the previous section, the goal of this section is to define the modeling basis for efficiently reusing situations, including their exposure assessment result. The sub-challenges in this section are to (define and) integrate the concept of "Exposure Assessment Result" while being constrained by the following two goals:

1. We aim at supporting the safety engineer in the assessment process.

   We therefore need to formalize the assessment, thus update our model.
2. Our overall approach SAHARA should be practicably applicable.

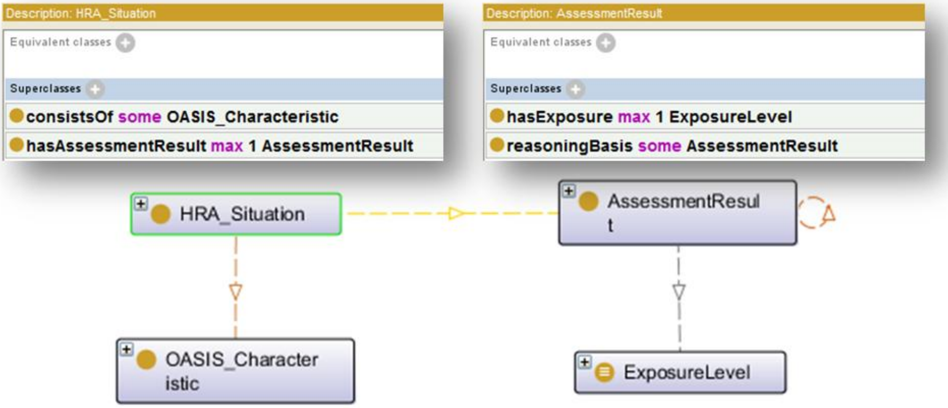
   We therefore need to consider practical requirements as well.

The status of our model is that we have the concept of influence factors and the concept of a situation. While keeping the above-mentioned constraints in mind, we need to add an exposure level to our concept of "situation". An easy and straightforward approach would be to introduce an attribute "exposure level" to a situation with possible values from E0 to E4. On the one hand, this would be a valid solution if our goal was to conserve only the exposure value as part of our formalization approach. However, due to the fact that our goal is to support the safety engineer in his assessment decision process, the attribute solution has two problems:

1. The exposure value is an implicit concept (as an attribute value) and therefore makes it hard to reason and automate things related to this domain concept. This is due to the fact that reasoning in an ontology is done via concepts and instances, not via attributes of instances. Thus, if we were to use as an attribute the constraint number 1 from above, a (semi-) automated assessment would not be possible or we would have to use a technical workaround.
2. As described above, the determination of an exposure value is a decision process in which the safety engineer uses and combines information. Reusing assessment results essentially means reusing the decision process. By introducing "only" an attribute, we cannot capture the decision process appropriately. We need the assessment, or better yet, the assessment result as an explicit concept that we can equip with information about the decision process. This is also a demand of the practical applicability constraint introduced above, because one piece of information that is highly relevant in practice is who came up with the assessment result. Having a concept for that allows us to add attributes to this concept, such as "CompanyName" and "SafetyEngineer".

Our solution for extending the model is therefore the introduction of two new concepts: "*ExposureLevel*" and "*AssessmentResult*". The "*AssessmentResult*" plays the role of connecting an *HRA_Situation* to an *ExposureLevel*, which is the basic requirement we had from a technical HRA viewpoint.

Figure 3.30 (partially) visualizes the realization in our ontology. In the upper part of the figure, one can see the textual definition of the relations introduced above as axiomatic restrictions on the respective concepts. Note that the upper left part defines the relations of the class *AssessmentResult*, while the right upper part defines the ones from *HRA_situation*. For software engineers who are used to (meta-) modeling in the UML, the way the relations are defined here might seem strange. However, this is how relations between concepts are defined in an ontology; the equivalent modeling concept in UML would be to define an association. To be precise, the concept used here is only similar and can be used for our purpose. What is defined in the ontology are, in fact, anonymous superclasses that the concrete classes have as parents! An *HRA_Situation* is, for example, the child of an unnamed class to which all those instances belong that have some *OASIS_Characteristic* defined in their *consistsOf* relation. Thus the ontology describes things and relations and states how things are and not how things can be built (cf. discussion on meta-modeling vs. ontology engineering).



Description: HRA_Situation

Equivalent classes

Superclasses

consistsOf some OASIS_Characteristic

hasAssessmentResult max 1 AssessmentResult

Description: AssessmentResult

Equivalent classes

Superclasses

hasExposure max 1 ExposureLevel

reasoningBasis some AssessmentResult

Figure 3.30: OASIS Characteristics extension

Corresponding to the requirements discussed above, the concept of *AssessmentResult* serves the following two purposes:

1. **Keeping track of the assessment decision:**
   AssessmentResult has a relation (called reasoningBasis) to itself (AssessmentResult). That allows us to store the information that other AssessmentResults were used in the decision process that resulted in another AssessmentResult. If we provide decision support by presenting similar situations to the safety engineer, it is interesting to know which of the presented previous assessments he used. He can mark those and our relation "reasoningBasis" stores this information.

   Another important aspect for both practicability and scientifically is the maintenance of situation assessments. Over time, situations may be assessed differently because new information is available or just because an old assessment was proven to be incorrect. If the updated information is used, one should additionally know which situation assessments were based on the previous result. This information is important because it might need to be reassessed as well. In safety engineering, an impact analysis has to be performed for each change of safety-relevant elements. This tracking and possible re-assessment is exactly such an impact analysis scenario. With our (simple) relation "reasoningBasis" we can trace the decision other, process and automatically derive possibly affected assessments.

2. **Making our model fit for practice:**

   This class makes it possible to perform future extensions of the assessment. In an industrial setting and usage of this ontology, it is most certainly interesting to know who assessed the situation. Thus, having the concept "AssessmentResult" makes it possible to further refine what assessment-relevant information is. Even switches to exclude certain assessment results for use in a product line context would be possible.

   While we have multiple ideas about which information (of an assessment) could be additionally relevant in *practice*, we want to focus on the core process of an HRA and introduce only scientifically relevant elements. We therefore omitted (for example) the definition of the simple attribute containing the name of the safety expert who performed the assessment.

In order to make the usage of this model more tangible, we will discuss a small example in the following. Let's assume we want to assess a situation that is informally described as:

"$\underline{\text{Overtake}}$ on a $\underline{\text{highway}}$ with a $\underline{\text{velocity of more than } 130km/h}$ and an acceleration of $\underline{+3m/s^2}$"

The OASIS influence factors are underlined in the informal description. Extracting the OASIS ontology fraction of this definition results in Figure 3.31.

The next step in this example would be to assess the situation and assign an exposure value as the assessment result. We assume that the assessment result is the above mentioned and modeled situation (*Situation1*), **E3**. Figure 3.32 is showing the modeling of *Situation1* along with two more examples of similar situations (*HighwaySituation*, *HighwayOvertakeSituation*) along with their *AssessmentResult*s (*HighwaySituationAssessment*, *HighwayOvertakeSituationAssessment*). The assessment results were used in this example (as a reasoning basis) to justify the exposure assessment of *Situation1*. This is visible by the dotted lines that are marked by a red rectangle, from *Situation1Assessment* to the other two AssessmentResult instances: *HighwaySituationAssessment* and *HighwayOvertakeSituationAssessment*.

We conclude this by summarizing that we have presented an extension of our ontology model for integrating an *AssessmentResult*, including the respective *ExposureLevel*, into the model. This step therefore gives us the

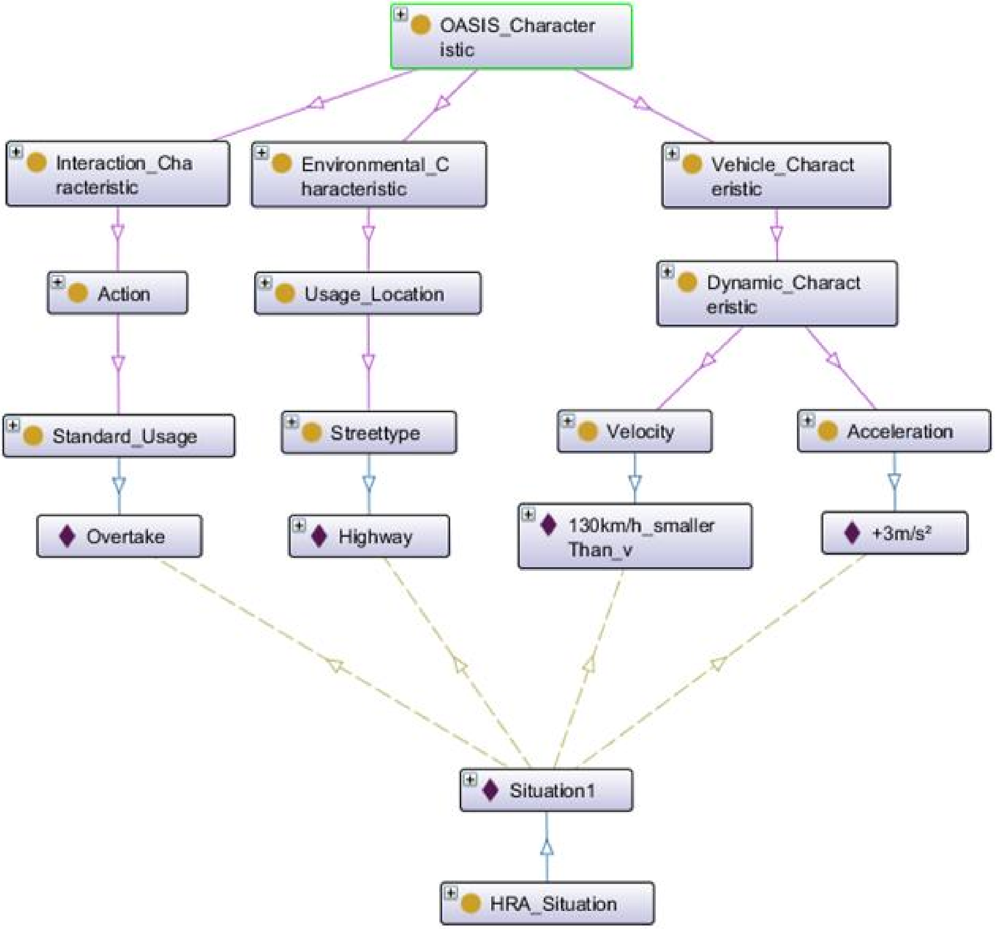


Figure 3.31: (Internal) Situation representation example

basis for working with HRA situations, including their exposure assessment, in a more formal way. The reason for the decision to introduce a concept *AssessmentResult* rather than "only" an attribute was (1) to formally store which (former) AssessmentResults were used in another assessment and (2) to enable better extensibility by other attributes that might be relevant in an industrial setting.

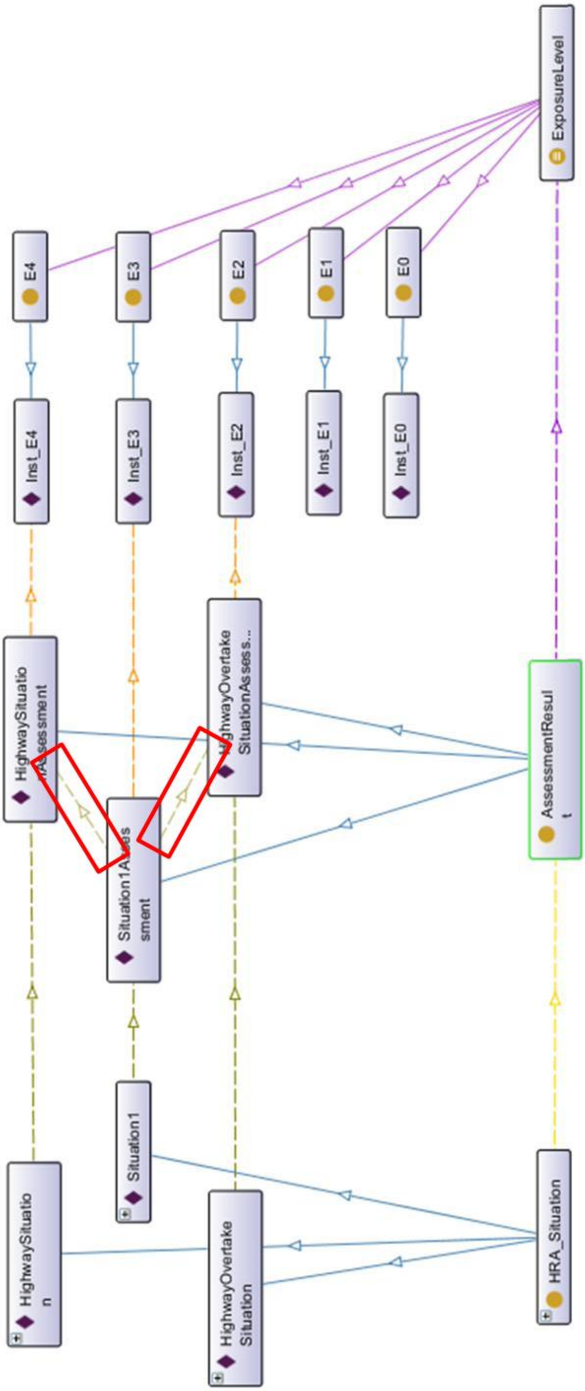


Figure 3.32: (Internal) assessment result example

3.4.2 HRA Situation similarity

In Figure 3.32 of the previous section, we have already seen the possibility of tracing former situation assessments back to the current assessment if those were used as a reasoning basis. The assumption is that the situations are similar enough to be relevant for the current assessment.

In this section, we will therefore discuss how to define HRA situation similarity more formally. We approach this by first discussing instance similarity in an ontology. This means we will discuss means for calculating the similarity of two instances in an ontology. In previous sections, we stated that an HRA situation is basically a set of influence factors. Thus, after discussing general instance similarity concepts, we will transfer this idea to similarity of sets, which will then give us our HRA situation similarity function.

The challenge with similarity is that everyone has his own (natural) opinion on similarity. Thus, if we ask an expert whether two HRA situations are similar, he can give us an answer, but the exact formalization of the similarity decision process is quite difficult.

If we do, for example, change "Situation1" from above (cf. Figure 3.31) and change the vehicle speed to a lower value such as "between $100km/h$ and $130km/h$", everyone would say the new situation is very similar. The reason for this is that the structure (the building blocks) did not change a lot. What we did is to move one influence factor instance to another sibling of the same parent (same concept: velocity). Thus, we have structural (or topological) similarity in the ontology.

The Requirements

Before discussing our similarity formalization approach, we will first present the relevant requirements an HRA situation similarity function has to fulfill.

Similarity will be used to support the safety engineer in the decision process for finding an exposure value for a new HRA situation. What we therefore need is a function that is as close to the expectation the safety engineer has regarding “similarity” of HRA situations. For an expert, a situation A is considered to be different from situation B if the picture he has in mind if he thinks of situation B has nothing to do with the picture he has in mind for situation A. It is similar if only slight changes of the picture occur.

More formally speaking, the mental (dis)similarity of situations is a check of how many influence factors change and how severe the change is. Change severity can be formalized by its commonalities. In vernacular language, the idiom “to compare apples with oranges”, for example, means comparing dissimilar things. Well, apples are not oranges, but they are both fruits; thus, they have a commonality. If we were to compare apples and tires, we would judge them as even more dissimilar. One commonality one could identify is that both are physical objects.

The same is true for influence factor similarity and therefore for HRA situation similarity. If the common covering concept is very abstract, one judges the two influence factors under consideration as different; if they share a closely related concept, they are (considered) similar. If, for example, we compare "heavy rain" and "one-way street", it is very hard to find a commonality; thus, these two influence factors are very different. However, if we compare "$> 130km/h$" to "$100km/h <= v < 130km/h$" (as discussed above), the common concept is vehicle velocity, which is very close, very concrete. Thus everyone would say that these influence factors are very similar. Based on these considerations we can derive two (high-level) requirements for our HRA situation similarity function:

1. **Reflect generalization/specialization:** If a situation *A* is a generalization of another situation *B*, this shall be adequately reflected in our similarity function. The generalization/specialization relation denotes whether one situation *A* is covered by another situation *B*. All influence factors of *A* would be in *B*. In such a situation, the exposure value of *B* cannot be higher than the one of *A* because each additional influence factor further restricts (the likelihood of) a situation. This should be reflected in the result of our similarity function: Comparing a general situation to a more specific situation, the similarity value should be negative (because the exposure value would be reduced in this case). Figure 3.33 shows an example of generalization/specialization.

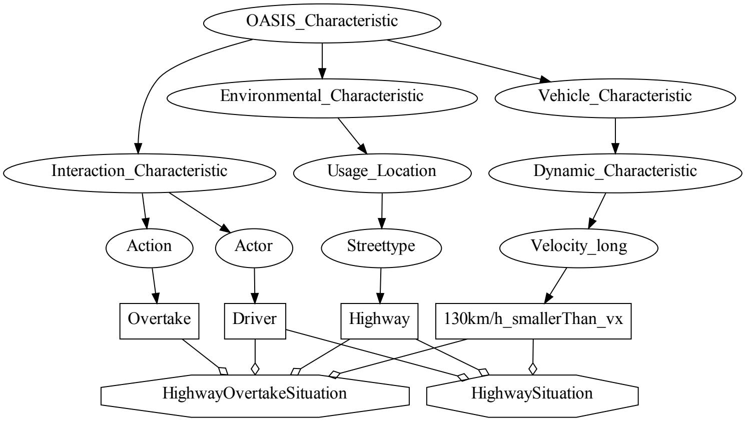


Figure 3.33: Example of generalization/specialization of HRA situations

At the bottom of the figure, two situations are defined: *HighwaySituation* and *HighwayOvertakeSituation*. The latter is a specialization of the former: *HighwayOvertakeSituation* contains all influence factors *HighwaySituation* contains, but in addition it has the influence factor *Overtake*.

2. **Change robustness:** Due to the fact that our ontology is subject to continuous extensions, we have to deal with these changes in the similarity metric as well. If a change is made in the ontology, the similarity metric of two situations should only change if the concept hierarchy the influence factors of the situation belong to changes. This might happen if the concept hierarchy is further refined, meaning an intermediate concept is added. Figure 3.34 shows an example illustrating this requirement.

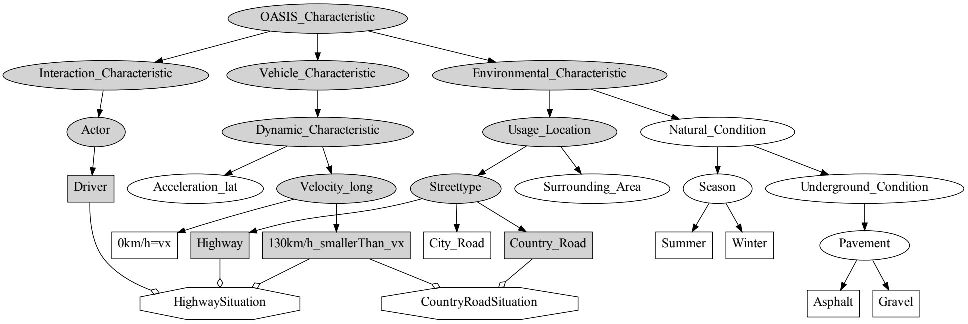


Figure 3.34: Example of change robustness of HRA situation similarity

The example shows two situations: *CountryRoadSituation* and *HighwaySituation*. Concepts and influence factors that are directly (and indirectly) related to these situations have a gray background. If the concept or influence factor is independent of the situations, it has a white background. The expectation of (and the requirement for) a similarity function calculating the similarity of the two situations is that it should depend only on the structure of the elements with the gray background. If, for example, the sub-tree on the right *Natural_ Condition* changes, one would not expect the similarity value between *CountryRoadSituation* and *HighwaySituation* to change.

These general requirements will guide us in the process of our HRA situation similarity discussion. Note that we have another "soft" requirement or rather a constraint: Our goal should be to represent/reflect the expert's expectations.

Foundation and theoretical discussion

We begin our discussion on situation similarity by reviewing Recio-Garcia's approach to structural similarity in ontologies [RgDaGc^{+}06] and discuss the applicability of his approach for our goal: HRA situation similarity. Garcia has proposed different ontology-based distance metrics that are based on the location of the element (instance) in the ontology. Due to the fact that Recio-Garcia's work is based on instance similarity and our first requirement

(Generalization/Specialization) is only applicable to sets of influence factors, we will review the functions with respect to the second requirement (and our constraint to be as close to experts' expectations as possible). In order to understand Recio-Garcia's instance similarity functions, he introduces some helper functions, which we use as well:

$super_C(c, C)$ is the subset of concepts in C that are superconcepts of c

$super_i(i, C)$ is the subset of concepts in C of which i is an instance

CN is the set of all concepts in the current knowledge base

$LCS(i_1, i_2)$ is the set of the least common subsumer concepts of the two individuals. Note that a "least common subsumer" concept is a concept both instances belong to, but the concept has no child-concept to which both instances belong! (Very informally, it can be described as the first common concept.)

$prof(Ci)$ is the depth/profundity of the concept Ci. Note that the "depth" is the graph distance of a node to the root.

$prof(i)$ is the depth/profundity of the individual i

With these helper functions, Recio-Garcia defined the following three similarity functions. All of these functions have as domain a pair of ontology instances and as range a real number between [0, 1], where 0 indicates no similarity and 1 total similarity.

$$fdeep_{basic}(i_1, i_2) = \frac{max\{prof(LCS(i_1, i_2))\}}{max\{prof(Ci)\}, \forall Ci \in CN} \tag{3.2}$$

The first similarity function (3.2) calculates the relation between the least common subsumer concept and the maximal concept depth in the ontology. Let's look at an abstract example (cf. Figure 3.35) to better understand this similarity function.

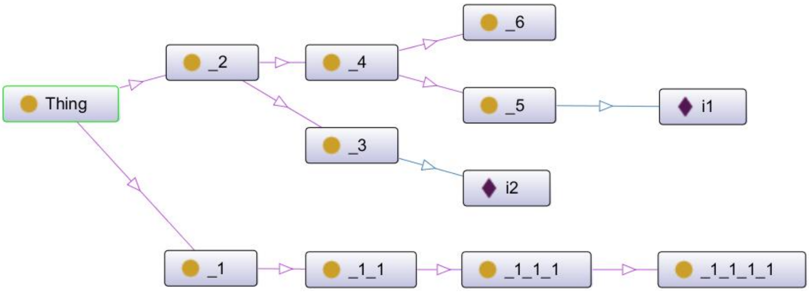


Figure 3.35: Abstract ontology similarity example

We are interested in the similarity between *i1* and *i2*. The numerator says:

$max\left\{prof\left\{LCS\left(i1,i2\right)\right\}\right\}$

The least-common concept (LCS) of *i1* and *i2* is the concept "_ *2*". This is the first node the two instances have in common if we go along the path from the instance back to the root ("*Thing*"). We get:

$max\left\{prof\left\{"\_2"\right\}\right\} = 1$

The profundity ("prof") of concept "_ *2*" is *1*. (This is essentially the graph distance of a node to the root.) Due to the fact that our LCS set has only one element (because we have a tree structure), the max operator returns the value of prof("_ *2*"), which is *1*.

The analysis of the formula yields the finding that the denominator is interesting: This is because it includes all other concepts/classes existing in the ontology. It asks for the maximal profundity (depth) of concepts in the ontology/graph. In our example, the concept with the highest profundity value is "_ *1*_ *1*_ *1*_ *1*":

$prof("\_1\_1\_1\_1") = 4$, thus,
$fdeep_{basic}(i1,i2) = 1/4 = 0.25$

The conclusion of the review of this distance metric is that this similarity metric is not change-robust (does not fulfill our requirement 2): If the path starting with _ *1* changes in a way that the maximal concept profundity of this branch changes, the similarity of *i1* and *i2* changes as well.

Thus, this similarity metric is not appropriate for our purpose because one could not, for example, calculate the similarity of our vehicle speed instances (from the example above) without knowing the complete remaining ontology.

Garcia provides another version that is shown in (3.3). This similarity function has the same numerator as Equation 3.2, but the denominator focuses only on the local sub-ontologies of concepts to which the respective instances *i1* and *i2* belong.

$$fdeep\left(i_1,i_2\right) = \frac{max\left\{prof\left(LCS\left(i_1,i_2\right)\right)\right\}}{max\left\{prof\left(i_1\right),prof\left(i_2\right)\right\}} \tag{3.3}$$

Informally speaking, this similarity function calculates the relative depth/height of the least common concept and the (maximum) depth of the individuals. For this it uses only information given by the instances (and their parent concepts). Therefore, it fulfills our requirement of change robustness.

The result of *fdeep* applied to our example (cf. Figure 3.35) is:

$$fdeep\left(i1, i2\right) = 1/max\left\{4, 3\right\} = \frac{1}{4}$$

However, one question remaining is whether our constraint about the expert's expectation is met. In order to elaborate on this we introduce yet another instance similarity function (of Recio-Garcia) (cf. Figure 3.4):

$$cosine\left(i_1, i_2\right) = \frac{\left|super_i\left(i_1, CN\right) \cap super_i\left(i_2, CN\right)\right|}{\sqrt{\left|super_i\left(i_1, CN\right)\right|} * \sqrt{\left|super_i\left(i_2, CN\right)\right|}} \tag{3.4}$$

This function may look complex, but it is in fact quite easy to understand: In the numerator, the number of common concepts of the two instances is calculated. This is divided by the geometric mean of the number of concepts belonging to each of the instances. The geometric mean serves the purpose of “normalizing” the number of concepts. Due to the fact that in our ontology, one influence factor belongs to exactly one concept in the ontology, we can replace the numerator by the depth/height of the least common concept of the respective instances. However, we have already seen this as the numerator of the two similarity functions $fdeep_{basic}$ and $fdeep$ discussed above! In this case, the denominator is in this case the geometric mean of the depths of the instances. Equation 3.5 shows "our" version of the similarity function as discussed:

$$cosine_{SAHARA}\left(i_1, i_2\right) = \frac{prof\left(LCS\left(i_1, i_2\right)\right)}{\sqrt{\left|super_i\left(i_1, CN\right)\right|} * \sqrt{\left|super_i\left(i_2, CN\right)\right|}} \tag{3.5}$$

Thus, the difference is that $cosine$ (and $cosine_{SAHARA}$) does not divide the numerator by the maximum of the depths of the instances (like in $fdeep$), but combines the depths of both instances. This satisfies our constraint for alignment with experts expectations better. This can easily be seen in the following example, shown in Figure 3.36.

$$fdeep\left(iAAAA, iBBBB\right) = \quad 1/max\left\{5, 5\right\} = \quad 1/5 \tag{3.6}$$

$$fdeep\left(iAAAA, iB\right) = \quad 1/max\left\{5, 2\right\} = \quad 1/5 \tag{3.7}$$

Even though iBBBB is "further away", the similarity calculated with $fdeep$ yields the same result for both comparisons.

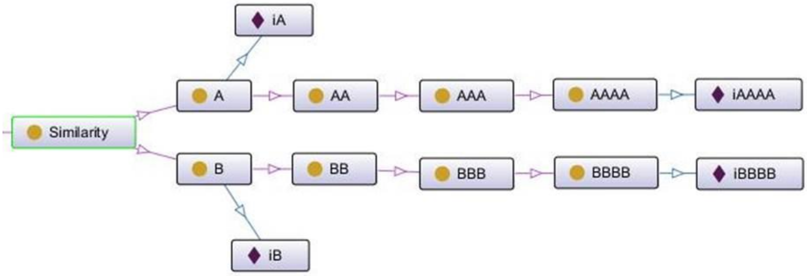


Figure 3.36: Example of fairness of $cosine$ and $fdeep$

In contrast, if we use $cosine$ as a similarity function, we get:

$$cosine\,(iAAAA, iBBBB) = 1/\left(\sqrt{5} * \sqrt{5}\right) = 1/5 = 0.2 \quad (3.8)$$

$$cosine\,(iAAAA, iB) = 1/\left(\sqrt{5} * \sqrt{2}\right) = 1/\sqrt{10} \approx 0.316 \quad (3.9)$$

The path from $iAAAA \rightarrow iBBBB$ is longer and has a similarity value of 0.2. The shorter path $iAAAA \rightarrow iB$ has a value of (around) 0.32, which implies higher similarity.

Thus, the similarity function $cosine$ is closer to the natural interpretation of similarity. This is true because $fdeep$ strictly "selects" the concept hierarchy of *one* instance and does not respect the other instance. In this example, the selection process is given by the "max" function. Cosine, in contrast, uses the geometric mean. By doing so, it incorporates the concept hierarchies of both influence factors.

We conclude our review of Recio-Garcia's ontology (instance) similarity functions by stating two general findings:

- The construct "Least-Common-Subsumer-Concept" gives us a partitioning point for the ontology. All concepts towards the root are contained in both influence factor concept hierarchies; thus, they contribute to the notion of "similarity". The "dissimilarity", in contrast, is induced by the structure beyond the LCS concept (towards the instances).
- One has to be careful with strictly selective functions, such as "max", because they allow only one concept to "contribute" to the similarity result. The expectation of similarity (in the domain of functional safety and HRA situations) usually incorporates information about both concepts for coming up with a similarity assessment. Thus, the danger of selective functions is that one loses the "natural" expectation. Normalizations with a geometric mean are more suitable than the maximum "mean".

With $fdeep$ and $cosine$, Recio-Garcia proposed two similarity functions; one uses the maximum as mean, the other one uses the geometric mean. These are, however, just two means, and there exist many more. The question is if there is another mean that suits our needs even better.

Discussion of Hölder mean
For a more in-depth analysis of possible means we used a generalization of all (power) means, the Hölder mean. (The Hölder mean is sometimes also termed “generalized mean”.) Equation 3.10 shows the specification of the Hölder mean:

$$M_p\left(x_1, \ldots, x_n\right) = \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i^p\right)^{1/p} = \sqrt[p]{\frac{x_1^p + \ldots + x_n^p}{n}} \tag{3.10}$$

Many “common” means are (or can be expressed as) special cases of the Hölder mean. Table 3.2 shows some of the well-known means expressed as instances of the Hölder mean:

Table 3.2: Special cases of the Hölder mean

Short-name	Mathematical definition
min	$\lim_{p\to-\infty} M_p\left(x_1, \ldots, x_n\right) = min\left\{x_1, \ldots, x_n\right\}$
$\bar{x}_{harm}$	$M_{-1}\left(x_1, \ldots, x_n\right) = \frac{n}{\frac{1}{x_1}+\ldots+\frac{1}{x_n}}$
$\bar{x}_{geom}$	$\lim_{p\to 0} M_p\left(x_1, \ldots, x_n\right) = \sqrt[n]{x_1 \cdot \ldots \cdot x_n}$
$\bar{x}_{arithm}$	$M_1\left(x_1, \ldots, x_n\right) = \frac{x_1+\ldots+x_n}{n}$
$\bar{x}_{rsm}$	$M_2\left(x_1, \ldots, x_n\right) = \sqrt{\frac{x_1^2+\ldots+x_n^2}{n}}$
max	$\lim_{p\to\infty} M_p\left(x_1, \ldots, x_n\right) = max\left\{x_1, \ldots, x_n\right\}$

Cauchy’s proof of "inequality of arithmetic and geometric mean" was further generalized using the Hölder mean:

$$min\left(x_1, \ldots, x_n\right) \leq \bar{x}_{harm} \leq \bar{x}_{geom} \leq \bar{x}_{arithm} \leq \bar{x}_{rsm} \leq max\left(x_1, \ldots, x_n\right) \tag{3.11}$$

This chain of inequations shows that increasing the order of power (of the Hölder mean) always results in a value that is at least as high as the values produced by all lower order means. For our purpose of HRA situation similarity, we use a mean for normalizations, i.e., as the denominator. As the denominator, we want a mean that does not result in being zero. In this sense, the maximum is of course a good function, but, as discussed above, it is too "selective". The geometric mean, on the other hand, delivers "0" if one of the elements is equal to "0". Based on the "inequality chain" above, we can derive that there is a mean with $p > 0$ that serves our needs. By analyzing the special cases shown above, we could derive that the arithmetic mean (p=1) is already valid (wrt. our requirements). Figure 3.37 shows the arithmetic (p=1), the rooted-square mean (p=2), and a higher-order power mean (p=4) for two values ranging both from 0 to 10. Please note that the x-axis (to the left) and the y-axis (into the page) represent the two values and the z-axis (upwards) shows the corresponding mean value.

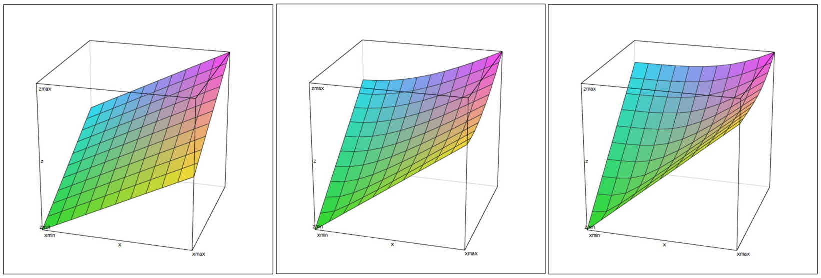


Figure 3.37: Example means: arithmetic (p=1), rooted-square (p=2), order-4 (p=4)

Due to the nature of (power) means, they deliver the same result if both values are the same! The difference is the way the mean "behaves" if the two values are different. The observation is that the higher-order power means favor high values in the sense that the impact a low value has becomes more and more unimportant. The extreme case is $\lim_{p\to\infty} = max$, where only the maximum value is used. Favoring high values and emphasizing high difference values is for the purpose of HRA situation similarity is not necessary and could not be justified; thus, lower-order means such as the arithmetic or rooted-square mean are appropriate for us.

Based on this review of means, we can update our $cosine_{SAHARA}$ similarity function and use the arithmetic mean instead of the geometric mean:

$$instanceSim_{SAHARA}\left(i_1, i_2\right) = \frac{prof\left(LCS\left(i_1, i_2\right)\right)}{\frac{|super_i(i_1,CN)|+|super_i(i_2,CN)|}{2}} \tag{3.12}$$

Up to now, we have discussed similarity metrics in general and derived some basic principles for defining our HRA situation similarity function. In the discussion, we focused on similarity functions working on ontologies (due to the review of Recio-Garcia's work), but all of the reviewed similarity functions were defined based on the instance level. While an HRA situation is an instance as well, similarity is defined by its (referenced) building blocks: the set of influence factors. In the next section, we will therefore transfer our lessons learned from single influence factor similarity to similarity of sets of influence factors. This will enable us to deal with the requirement for "Generalization/Specialization" as well.

The transfer towards HRA situation similarity

In this section, we will present our approach for calculating the similarity of HRA situations. For this we combine our fundamental and theoretical findings from the previous chapters in order to come up with a similarity function that fulfills our requirements introduced above.

The general structure of this section and of our solution is as follows:

1. We extend the notion of similarity from (single) influence factor similarity to set similarity (because an HRA situation is a set of influence factors).
2. Then we transfer the idea of the Least-Common-Subsumer concept to sets of influence factors. As discussed above, the LCS concept divides the ontology into similar and dissimilar parts. We use this partition for fulfilling the change robustness requirement.
3. We introduce a metric that incorporates the generalization/specialization requirement.
4. We tweak our solution based on practical experience to better fulfill the constraint of being as close to HRA experts notion of similarity as possible.

(1) The naïve approach towards set similarity is to calculate the similarity of all pairs of elements and combine these into a set similarity. While the general idea is straightforward and correct, the challenge here is to decide which pairs of influence factors to use for the similarity calculation. A set is unordered by definition. There is no induced way to pair up elements from two sets. A typical approach is, of course, to form the “cross-product”, i.e., to create all possible pairs. However, this approach generates many "useless" pairs - "useless" in the sense of comparing apples and oranges. That one builds "useless" pairs is bad, but this approach would actually be even worse because the number of useless pairs would reduce the impact of the relevant pairs. The question is what a “relevant” pair is. What one expects is that we build pairs that are closely related, meaning pairs of influence factors that are similar. However, these are cyclic requirements in the sense that we need the similarity values of all pairs in order to decide which pairs to calculate the similarity for.

The way out of this dilemma is that to incorporate information that gives us domain knowledge on similarity: the concept hierarchy. Recall that the concept hierarchy was the basis for all similarity functions discussed above.

We need to look deeper into the characteristics of our ontology. Technically, our/an ontology is a (directed, acyclic) graph structure. The influence factors of an HRA situation are leaves or endpoints of this graph. The concept hierarchy to which the (selected) influence factors belong is a path in the graph from/towards the root. As discussed in the review of the influence factor similarity functions, the similarity of two influence factors can be calculated by some sort of relation between common path segments (of both influence factor paths) and overall path segments or even maximal paths in the ontology. The general pattern is/was:

$$SimilarityValue = \frac{SimilarFraction}{SimilarFraction + DissimilarFraction} \tag{3.13}$$

For the extension towards a set of influence factors we therefore need to define a similar part and dissimilar parts. The challenge is that we have to deal with multiple paths. Some segments of these paths are shared between two situations, whereas some segments are exclusive to one situation. The overall "Least-Common-Subsumer-Concept" is the influence factor root, $OASIS_{Characteristic}$. Thus, we have a common root of all paths. Due to the fact that all influence factors belong exactly to one path, the directed, acyclic graph is more precisely a tree structure in our case. HRA situation similarity can therefore (technically) be reduced to graph similarity of tree structures.

Given one HRA situation and the OASIS ontology, the tree representation of the HRA situation constitutes a sub-ontology, or a sub-graph of the overall OASIS ontology. In this tree, only such information is contained that is somehow related to the situation. The situation-induced tree structure already fulfills the "Change robustness" requirement because only direct or indirect information is contained in an HRA situation tree. Figure 3.34 shows a tree representation of the sub-ontology induced by the two situations *CountryRoadSituation* and *HighwaySituation*.

(2) As stated above, calculating a tree similarity is often done by calculating common parts and different parts. We already stated that we can leverage the LCS concept to distinguish between similar and dissimilar parts. If we consider the tree representation of the (two) HRA situations we want to compare, we can use the LCS concept on each path to divide the path into a common segment and potentially a different segment. The common path segment obviously starts at the root and continues up to the point where the child concepts are different (or only the influence factors are different). If we apply this to two trees, we can define the LCS of two trees to be the maximal sub-tree covered by both trees. This abstract sub-tree therefore defines our border between similarity and dissimilarity. Applied to our application domain, the HRA situation similarity, the abstract sub-tree defines an abstract (common) sub-situation. Note that "abstract" in this context means that the situation may contain branches ending in concepts, not in concrete influence factors. Figure 3.38 shows an example of the graph representation and the abstract (common) situation.

The example is already known from the change robustness example (cf. Figure 3.34. This time, however, the gray background does not indicate a direct or indirect dependency on *one* of the situations, but rather elements belonging to *both* situations. Thus, the abstract common situation is given by the conglomerate of the gray-backgrounded elements. An example of a common concept without a common influence factor is given by the concept *"Street Type"*.

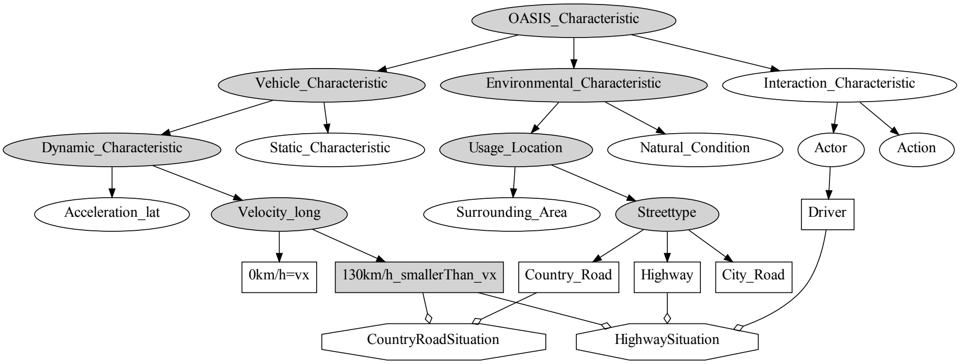


Figure 3.38: Example of abstract (common) HRA situation

In order to formalize an abstract situation, we define two helper functions:

$$InfSet\left(Situation_{OASIS}\right) := \{i \in InflFactSet_{OASIS} \mid i \in Situation_{OASIS}\} \tag{3.14}$$

$$ConSet'\left(i \in InflFactSet_{OASIS}\right) := \{c \in ConceptSet_{OASIS} \mid i \text{ instanceOf } c \vee \exists c' \in ConSet'(i) : i \text{ instanceOf } c' \wedge c' \text{ isSubclassOf } c\} \tag{3.15}$$

$$ConSet\left(Situation_{OASIS}\right) := \bigcup_{i \in InfSet\left(Situation_{OASIS}\right)} ConSet'(i) \tag{3.16}$$

(3.14): The function $InfSet$ is defined for an OASIS situation. The function extracts the set of influence factors from the OASIS situation. In this function, the set $InflFactSet_{OASIS}$ denotes in this function the set of all influence factors contained in the OASIS ontology.

(3.15): In contrast, the function $ConSet'$ calculates for one OASIS influence factor the concepts to which the influence factor belongs. (Analogously to $InflFactSet_{OASIS}$, $ConceptSet_{OASIS}$ denotes the set of all concepts in OASIS.) The definition is inductive: A concept is concept of an influence factor iff it is directly the parent of the influence factor, or if it is the parent of a concept which is a concept of the influence factor. With these helper functions we can define $abstSit\left(Sit1, Sit2\right)$, which calculates the abstract situation induced by *Sit1* and *Sit2* by:

(3.16): The function $ConSet$ combines both functions into a function calculating all concepts that are related to an OASIS situation.

$$comInfSet\left(Sit1, Sit2\right) := InfSet\left(Sit1\right) \cap InfSet\left(Sit2\right) \tag{3.17}$$

$$comConSet\left(Sit1, Sit2\right) := ConSet\left(Sit1\right) \cap ConSet\left(Sit2\right) \tag{3.18}$$

$$abstSit\left(Sit1, Sit2\right) := comInfSet\left(Sit1, Sit2\right) \cup comConSet\left(Sit1, Sit2\right) \tag{3.19}$$

(3.17): The function $comInfSet$ calculates the set of common influence factors of both situations. It is simply the intersection of the $InfSet$'s of both situations.

(3.18): The function $comConSet$ analogously calculates the set of common concepts.

(3.19): Finally, the function $absSit$ finally calculates the abstract situation. The abstract situation constitutes the commonality of both situations. It is therefore the union of the common influence factors (cf. $comInfSet$) and the common concepts (cf. $comConSet$).

With the concept of an abstract HRA situation, we can move forward towards the definition of our HRA situation similarity function. The very generic similarity function defined in equation (3.13) can be refined into:

$$HRASimFunc\left(Sit_{new}, Sit_{old}\right) = \frac{simSit}{simSit + mean\left(disSit_{new}, disSit_{old}\right)} \tag{3.20}$$

where

Sit_{new}, Sit_{old} : are the situations to be compared

$disSit_{new}$: is a value expressing the dissimilarity of Sit_{new}

$disSit_{old}$: is a value expressing the dissimilarity of Sit_{old}

$simSit$: is a value representing the similarity of Sit_{new}, Sit_{old}

$mean\left(\right)$: is a function calculating a/the mean

In order to make this similarity function "formal", we need to define how to calculate $disSit_{new}$, $disSit_{old}$, $simSit$, and we need to select a "mean".

The concept of abstract HRA situation introduced above is a tree. The commonality gets stronger the further away a common concept is from the root. In set theory, a tree is a partially ordered set $(T, <)$, with the relation "$<$" such that for each $t \in T$, the set $\{k \in T : k < t\}$ is well ordered. For each element t ($t \in T$), the order type of the element is called height of t, denoted $height\left(t, T\right)$. Informally speaking, the height of a tree node is the length of the path from the node to the root. (The root therefore has the height 0.)

For the calculation of the tree size we therefore attribute each node with its height. Let the tree graph be $G_{Tree} = (V, E)$, then we can calculate for each $v \in V$ the corresponding $height\,(v, G_{Tree})$, and with that we define:

$$treeSize\,(G_{Tree}) = \frac{\sum_{v \in V} height\,(v, G_{Tree})}{|V|} \tag{3.21}$$

With the $treeSize$ function we now can calculate "simSit" by using our $abstSit$ function defined above:

$$simSit = treeSize\,(abstSit\,(Sit_{new}, Sit_{old})) \tag{3.22}$$

$disSit1$ and $disSit2$ both represent values for the dissimilar part of the respective situations. Assume we have two situations Sit_{new} and Sit_{old} and their abstract situation tree. If we now “subtract” the common part of both situations from one of the situations, what is left is in most cases an unconnected graph. Figure 3.39 shows an example.

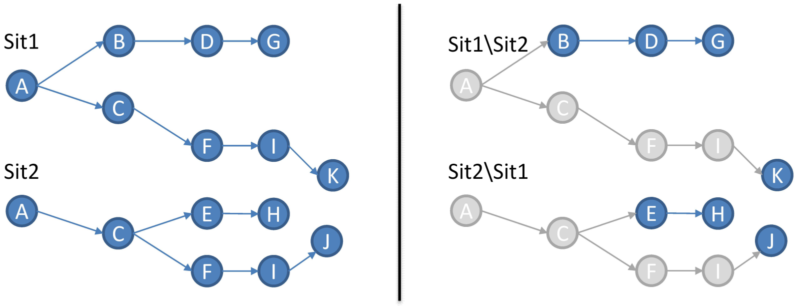


Figure 3.39: Example showing the dissimilar parts of two trees

On the left, both situations are schematically shown; on the right, the dissimilar parts are shown in blue, and the abstract situation (the common part) is shown in light gray.

In this example, the dissimilar part of Sit1, i.e., Sit1\Sit2, consists of the connected graph “$B \rightarrow D \rightarrow G$” and the single node “K”. Sit2\Sit1 analogously consists of “$E \rightarrow H$” and “J”.

The dissimilarity is higher the longer the paths or branches are. The dissimilarities “K” and “J” from the example are not as severe as the dissimilarities “$B \rightarrow D \rightarrow G$” and “$E \rightarrow H$”. The bigger a sub-graph is in the dissimilarity graph, the less similar our situations are. Due to the fact that the dissimilar part is in the denominator of our (informal) equation (3.20), higher sub-graphs need to get higher values. This is already the approach of our treeSize function! Due to the fact that we do not have one tree but possibly multiple tree branches, we introduce a virtual new dissimilarity-“root” and place the branches directly on the virtual root. In this way, we can use our $treeSize$ function as defined above to calculate the dissimilarities as well. Furthermore, let us define the helper function $disTree\,(Sit1, Sit2)$, which calculates the dissimilarity tree (including the virtual node).

Figure 3.40 shows abstractSituation, disSim1, and disSim2, each annotated with their *treeSize* value:

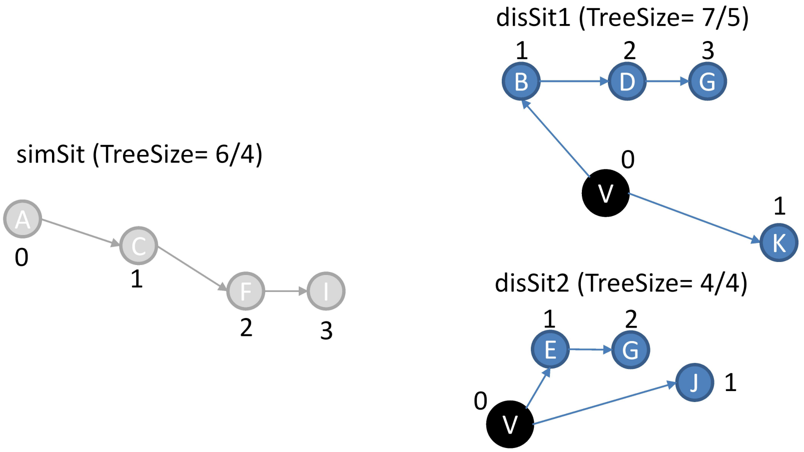


Figure 3.40: Example showing the basic similarity metric of two HRA trees

Now, we can almost construct our HRA situation similarity function. The missing piece is the "selection" of our mean. Due to our discussion above, we selected the arithmetic mean.

The (for now) final HRA situation similarity function is therefore defined as:

$$HRASimFunc\,(Sit_{new}, Sit_{old}) = \frac{simSit}{simSit + \frac{disSit_{new} + disSit_{old}}{2}} \tag{3.23}$$

where

$$\begin{aligned} disSit_{new} &= treeSize\,(disTree\,(Sit_{new}, Sit_{old})) \\ disSit_{old} &= treeSize\,(disTree\,(Sit_{old}, Sit_{new})) \\ simSit &= treeSize\,(abstSit\,(Sit_{new}, Sit_{old})) \end{aligned}$$

(3) While we can already calculate a similarity value, it does not (yet) reflect the aspect of *Generalization/Specialization* of the situations. With Generalization/Specialization we want to introduce a partial-order relation for situations. "Partially" for our situations is that the we can compare arbitrary situations, but we can only define a definite relation of being "bigger" or "smaller" in the case where one situation is contained in another situation. In this case the dissimilarity values between the covered situation and the covering situation is "0" because the *disTree* consists only of the introduced "virtual root", which has $treeSize = 0$. Due to the fact that the covering situation is "bigger", its dissimilarity tree contains more nodes than only the "virtual root" and therefore the value is > 0. If a situation is "greater" than another, it contains more influence factors. This makes the situation more unlikely, i.e., lowers the exposure value. We therefore want the similarity value to reflect a reduction: The similarity value should be negative. If both situations are equal, both disTree's would result in a dissimilarity value of "0", but we want the similarity to be one. We therefore introduce a factor "genSpeFact", which simply defines the algebraic sign for the *HRASimFunc*:

$$genSpeFact(Sit_{new}, Sit_{old}) = \begin{cases} 1 & \text{if } Sit_{new} \text{ is covered by } Sit_{old} \\ -1 & \text{if } Sit_{new} \text{ covers } Sit_{old} \\ 1 & \text{if } Sit_{new} \text{ is equal to } Sit_{old} \\ \in \{-1, 1\} & \text{else} \end{cases}$$

Note that in this definition we partitioned the comparison into being "greater or equal" or "smaller" because with case number 3, we unified the equality and the "greater" cases. Whether the exact similarity is positive or negative is (mathematically) unimportant. The only important aspect is that our $genSpeFact$ factor never becomes $= 0$.
The first two cases can easily be covered by a signum function:

$$genSpeFact(Sit_{new}, Sit_{old}) = sgn(disSit_{old} - disSit_{new}) \tag{3.24}$$

However, the problem is that the sgn function is defined to deliver "0" if the input is "0". Thus, case 3 is not yet fulfilled by equation (3.24). We can leverage the case that by definition $\frac{0}{0} = 1$. With that we can define:

$$genSpeFact(Sit_{new}, Sit_{old}) = \frac{(disSit_{old} - disSit_{new})}{|disSit_{old} - disSit_{new}|} \tag{3.25}$$

Now we can integrate the "genSpeFact" (3.25) into our HRASimFunc (3.23):

$$\begin{aligned} &HRASimFunc\left(Sit_{new}, Sit_{old}\right) = \\ &\quad genSpeFact(Sit_{new}, Sit_{old}) \cdot \frac{simSit}{simSit + \frac{disSit_{new}+disSit_{old}}{2}} \\ &\quad = \frac{(disSit_{old} - disSit_{new})}{|disSit_{old} - disSit_{new}|} \cdot \frac{simSit}{simSit + \frac{disSit_{new}+disSit_{old}}{2}} \end{aligned} \tag{3.26}$$

The Generalization/Specialization property of HRASimFunc is shown as an example in Figure 3.41.

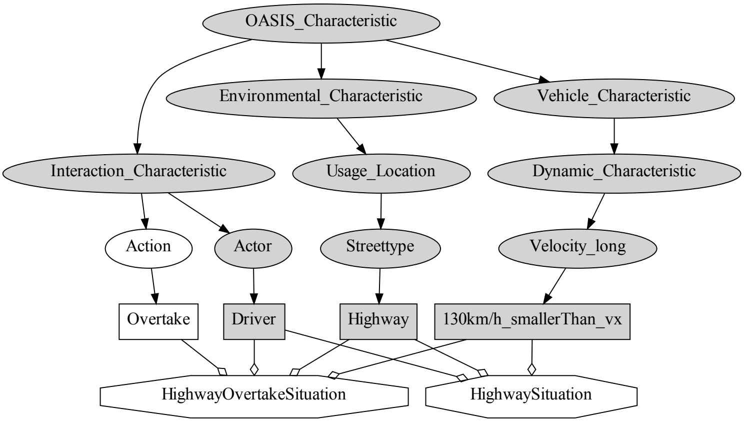


Figure 3.41: Example showing generalized/specialized situations

The example is the same as that introduced as part of the requirements discussion. This time we have colored the abstract situation gray. As we can see, the gray nodes completely cover the influence factors belonging to *HighwaySituation*. *HighwayOvertakeSituation* additionally has a maneuver, *"Overtake"*, specified and is therefore a special case of *HighwaySituation*!

The similarity function comparing *HighwaySituation* to *HighwayOvertakeSituation* delivers (step by step):

$$Sit_{new} = HighwaySituation \tag{3.27}$$
$$Sit_{old} = HighwayOvertakeSituation \tag{3.28}$$
$$disSit_{new} = 0 \tag{3.29}$$
$$disSit_{old} = 1 \tag{3.30}$$
$$simSit = 26/12 = 13/6 \tag{3.31}$$
$$\tag{3.32}$$

and

$$HRASimFunc\left(Sit_{new}, Sit_{old}\right) = \frac{(1-0)}{|1-0|} \cdot \frac{(13/6)}{(13/6) + \frac{0+1}{2}} = 0.8125 \tag{3.33}$$

If we switch the comparison and compare the old *HighwayOvertakeSituation* to the new *HighwaySituation*situation, we would get:

$$HRASimFunc\left(Sit_{old}, Sit_{new}\right) = \frac{(0-1)}{|0-1|} \cdot \tag{3.34}$$

The negative algebraic sign shows that given the assessment value (exposure) of the old situation, *HighwaySituation*, the exposure of the new situation, *HighwayOvertakeSituation*, must be lower! This complies with the expectation that an exposure assessment of a more specific situation results in lower values, while a generalized situation results in a higher exposure value.

(4) With the end of part (3) we have finalized our HRA-Situation similarity function. In evaluations with industry partners, however, we have found some issues that can be tweaked to better fulfill the industry's needs. We have identified two general improvements:

1. We often heard the comment/question:

   "The situation similarity is very nice, but can I provide a positive list or a negative list of influence factors that should be contained or shouldn't be contained in the similar situations?"

   The idea of just filtering the result with respect to a positive list and negative list of influence factors was not considered to be optimal, however, because we lose information. We "learned" that "should/shouldn't" is more of a soft requirement in the sense that the similarity value should be increased/decreased, but not that the situation per se should be removed.

2. Another fact we learned during an evaluation of our similarity value is that when we compared the experts' expectations with the values calculated, we found one interesting aspect: The top-level OASIS classes are kind of special - special in the sense that a difference in similarity that goes beyond the top-level class is expected to be "more" different than an (top-level) internal difference with the same similarity result. Thus, talking about vehicle dynamics and comparing this to environmental characteristics was considered to be more distinctive than a difference within environmental characteristics.

While theoretically not groundbreaking and more of an optimization issue, we decided to take care of these comments because our approach aims at practical applicability.

In the following section, we will therefore present our two improvements.

3.4.3 Exposure Assessment Support

As described at the beginning of section 3.4, the overall goal of HEAT is to increase the efficiency of reusing HRA situations. The direct benefit of reuse is, of course, to save time, money, and/or effort because one does not have to redo things. Another (desired) benefit is to increase the consistency of the product or system built with reused components. In our case we want to reuse domain knowledge in terms of assessments of situations. We see the following two benefits: (1) The direct benefit for a safety engineer is to speed up and get support for the assessment of new HRA situations. (2) The indirect benefit would be to make the exposure value of a situation less expert-dependent and ideally retrieve/have one exposure value for each situation, independent of the expert who is performing the assessment.

The Requirements

Based on the direct and indirect benefits mentioned above, we can derive two goals for this section: 1) support the engineer in the assessment task and 2) increase situation assessment consistency. The goals are not independent, of course. We should support the engineer in such a way that a new assessment is consistent with former assessments. The difference, however, is that 1) aims at the engineer and his *task* of performing an assessment, while 2) states a demand on the *artifacts*, the assessed situations.

Supporting the task of assessing a situation basically means supporting a (human) decision and reasoning process. Human decisions are a very complex field and still subject to research activities. Baron [Bar08] and Kahneman [KT72] list are around 100 effects and biases that influence and may ultimately alter decisions. It is not our goal to deal with all these influences and/or models of human decision making, but to obey basic observations and principles and derive assessment support based on these principles. The basic model is that a decision is influenced by external facts and internal knowledge, beliefs, or mental states. If something is decided in the same way by many persons, in psychology this is termed "intersubjective". If the decision is individual, it is termed "subjective". Transferring this to the assessment task of situations, we derive that every intersubjective result is an expert-independent assessment result and that a strongly expert-dependent one would be subjective. Thus, we need an approach that makes the assessment intersubjective. If we use the partitioning of decision influences into external and internal influences, we can state that the stronger the decision is based on external facts, the higher the likelihood that the decision is intersubjective. Our goal is therefore to support the engineer by providing good external facts for him to perform the assessment.

The other goal, consistent situation assessment, is per se important, but is also important for the task support discussed above. If the facts, the database, show an inconsistent picture, the expert might be confused and the intended support might result in a distraction/confusion instead. Thus, our consistency goal is to provide a way to check the consistency of a situation knowledge-base.

The concrete contributions we address in this chapter are therefore:

– We provide an approach that enables us to check a knowledge base for situation assessment consistency.

– We provide a concrete situation exposure assessment support approach.

The situation knowledge base consistency check

The stated principle of classical, intuitionistic and many other logic systems, "from a contradiction, anything follows", is of course immanently important for the assessment process because we do not want *anything* to be "follow"ed from our knowledge base. The fundamental question we need to answer first, however, is what a contradiction of a situation or an inconsistent HRA situation knowledge base is. As introduced above, a situation consists of a set of influence factors and an exposure value (assessment result). One single situation by itself cannot be inconsistent. Inconsistency is defined on a set of (or at least a pair of) situations. Please remember: an HRA situation consists of a set of influence factors and an exposure value. Both parts, the influence factor set and the exposure value, imply a kind of ordering relation on situations. We informally define that two (or more) situations are inconsistent if the ordering relation implied by the influence factor set is not the same as the one induced by the exposure value.

The goal of checking the consistency of an HRA situation knowledge base can therefore be realized by checking whether the two ordering relations do not contradict each other. Due to the importance of the ordering relations, we will recapitulate them briefly.

For the *set of influence factors*, we already discussed in the previous section the property of a situation to be a generalized/specialized situation:

A situation A is said to be more general than another situation B iff all influence factors of A are contained in B as well.

If we interpret being more general as "bigger" and more special as "smaller", we can obviously induce a partial order on a set of situations, with the ordering relation "general/special".

The *exposure ordering* relation is given by the total order of the integer numbers (or by the total order of the subset 0, 1, 2, 3, 4 used to define an exposure value).

The generalization/specialization relation induces a partial order while the exposure values define a total order. Checking for consistency therefore means that for each pair of situations having a generalization/specialization relation, the corresponding exposure values need to reflect this relation as well.

While the generalization/specialization of situations is already implemented in the HRA situation similarity function introduced above, we cannot use the function for the partial ordering. The reason is that the HRA situation similarity function is total (= it is defined for all pairs of situations) and we cannot extract the partial order induced by the generalization/specialization relation.

Therefore, we use only one part to define the HRA similarity function, the helper function InfSet (cf. equation (3.14)). This function gives us the set of influence factors belonging to a situation. The set inclusion gives us our partial order, which can easily be represented as an acyclic directed graph. Figure 3.42 shows a small example. On the right of the text "Situation" the exposure value is shown in parentheses: "Situation 2.2 (E3)" (shown in the middle) means that the situation 2.2 has the exposure value "3". Below each situation, we listed the set of influence factors that induce the partial order.

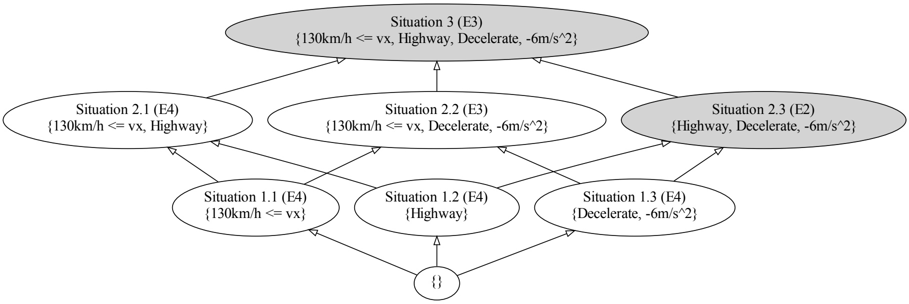


Figure 3.42: Consistency check partial order example

Checking the consistency of this graph figuratively means walking from the bottom (the empty situation/node) to the top node and checking whether the exposure values decrease or not. If a value decreases along the path, this would be an inconsistency. In the example, Situation 2.3 and 3 on the right and the top (nodes with gray background) constitute an inconsistency: Situation 3 is the successor of Situation 2.3, but the exposure value increases from 2 to 3 ($E2 \rightarrow E3$).

While this approach works and delivers correct results, the solution has the disadvantage that we need to duplicate the complete situation data in another tool (in memory). It is better to have an integrated solution, not only because of the data duplication, but also due to better integration into a holistic approach. Our solution is therefore to use the infrastructure and capabilities we have, and to let the ontology (or better the attached reasoner) do the work.

In order to leverage our ontology to do the work of ordering, we need to rewrite the situations slightly. The idea is to introduce for each situation (instance) a corresponding class. Note that we can distinguish between an instance and a class by the semantics that a class always describes a set, while an instance represents one individual. Turning an individual into a set essential means creating a set description in such a way that the set contains the individual. An abstract example is:

Let $i \in M$ be an individual. We can define $I := i \subset M$ as being the set induced by the individual.

If we now consider the fraction of a situation describing the set of influence factors, the "consistsOf" relation, we can create an equivalent class description for each situation similar to the abstract example shown above. Figure 3.43 shows the influence factor part as a "consistsOf" relation. (The figure is taken from the more complete example shown as Figure 3.31.)

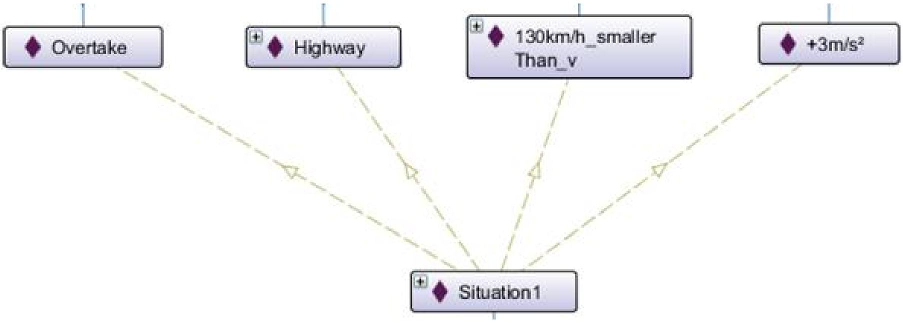


Figure 3.43: Situation instance representation example

With each influence factor we can describe a *set* of situations: the set of situations with a “consistsOf” relation to the influence factor. This means that we do not directly build class representations from the influence factors, but rather for the situation instances referencing the influence factor. If we create a class representation, a set, for each of the influence factors and create the intersection of these classes (or sets), we get a class (set) representation of the situation. To be more precise, we get a set describing all situations consisting of at least the set of influence factors. With "at least" we refer to the fact that situations are contained in the set that consists of more than the mentioned influence factors. The set "translation" example visualized as intersecting sets is shown in Figure 3.44 using a Venn diagram:

In terms of ontology engineering, we defined for each situation and equivalent class description. An example "translation" from individual representation to class representation is shown in Figure 3.45.

The reasoner uses the equivalent class representation an rebuilds a set hierarchy reflecting our desired partial order. This partial order can then be processed in terms of consistency checks. Due to the fact that the reasoner does not (only) aim directly at providing a partial order of sets, but tries to derive as much information about the sets as possible, we also get a list

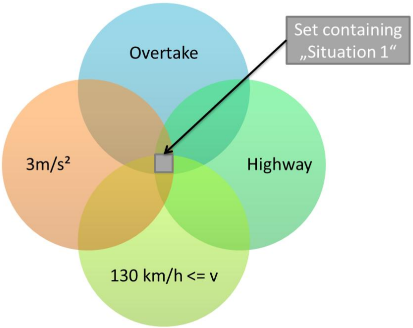
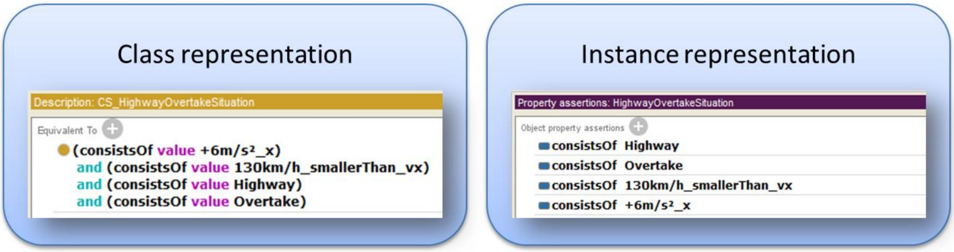


Figure 3.44: Venn diagram showing the class representation of an HRA situation



Class representation

Description: CS_HighwayOvertakeSituation

Equivalent To

● (consistsOf value +6m/s^2_x)
and (consistsOf value 130km/h_smallerThan_vx)
and (consistsOf value Highway)
and (consistsOf value Overtake)

Instance representation

Property assertions: HighwayOvertakeSituation

Object property assertions

consistsOf Highway
consistsOf Overtake
consistsOf 130km/h_smallerThan_vx
consistsOf +6m/s^2_x

Figure 3.45: Comparison of HRA situation representation as class and as individual

of equivalent situation as a "side effect". We implemented this as an HRA situation consistency check tool and applied it to industrial case studies. In the following, we present concrete results and statistics found with the situation check tool.

Table 3.3 shows the findings of a total of nine industrial case studies:

Table 3.3 shows in the top part general information about the situation knowledge base, while the lower part gives some more details about the inconsistency checking process.

The knowledge bases analyzed contained 148 situations, which we transferred "as is" from nine industrial case studies. We found that roughly 10% of the situations were defined (at least) twice. This means that if we had used the knowledge base as a simple database, without similarity suggestions, the assessment effort would have been reduced directly by 10%! Another finding we could derive as an additional "side effect" was that even though the situations came from industrial case studies, we found 12 situations without any exposure value, i.e., unassessed situations.

Table 3.3: Statistics of HRA situations based on 9 industry case studies

Parameter	Value
Overall number of situations	148
Duplicated situations	14
Situations without exposure value assigned	12
Exposure Inconsistencies	**19**
Partial-order chain lengths	4
PO with length 1	61
PO with length 2	44
PO with length 3	12
PO with length 4	4

The main area of interest in this section is the *consistency* of the knowledge base. With the proposed analysis approach, we found a total of 19 inconsistencies. This is a factor of roughly 13%. The case studies were performed under the umbrella of consistent and structured HRAs, which revealed the importance of consistency and made the experts focus on and pay attention to consistent assessments. A value of 13% inconsistency is already a high value. If we include the mental bias the experts had, one could assume that in daily work in industry, this value could even be higher. Note that the unassessed situations *neither* count as inconsistent *nor* do they influence the number of absolute inconsistencies. The percental inconsistency value, however, is decreased because unassessed situations are counted as "consistent".

Shown additionally in the lower part of the table are some details generated during the consistency check. Shown are the partial order chains. With partial order chains we denote the paths in the graph representation of the partial order induced by the influence factor partial order. The longest chain/path included four situations, and we had four of these chains. The other data can be interpreted analogously, i.e., we had 44 paths of length 2. This shows us that the situation hierarchy is very shallow (1.66 on average) and the lower the hierarchy, the lower the potential for situation inconsistencies.

We conclude this section by summarizing that we developed a theoretical and technical methodology for checking HRA situation consistency on the one hand and on the other hand leveraged the ontology and reasoning capabilities as an industry-strength technique for the implementation. Finally, we applied it practically to nine industrial case studies and were able to show that the theory as well as the applicability requirement are fulfilled.

Situation assessment support

In the last section, we presented a consistency check technique for our HRA situation knowledge base. For the part dealing with assessment support for safety experts, we assume that we applied the consistency technique and "improved" the knowledge base in such a way that the knowledge base is consistent. In this section, we will introduce our approach for supporting the assessment *task* by leveraging the HRA situation similarity function presented above and assuming a consistent knowledge base.

In the requirements part, we already discussed some basic background regarding the decision process of a situation exposure assessment. We stated that the goal is to find an "intersubjective" assessment result, meaning the result should be as independent as possible from an individual safety expert. In order to achieve this, we must deliver good external facts (as discussed above); "good" in the sense that they help the safety expert in the assessment.

During the design process of the decision support approach, we had to learn, however, that knowing which information is really helpful for the decision depends on the situation to be assessed, on the previous situations and assessments available (the knowledge captured so far), and to a large extent on the individual experts' opinions. We therefore created an approach that can be adapted to the needs of an expert. The "personalization" we created was to parameterize and filter the results we got by using our core approach, the application of the HRA situation similarity function. The concrete parameters/filters we realized are:

- *Set filtering*

  We provide the possibility to define sets of influence factors that the expert definitely wants to have in similar situations and a set that he does not want. An informal example is that an expert wants all similar situations that contain "Highway" but do not contain "driving straight". This means that he provides a positive and/or negative list of influence factors.
- *Restricted number of dissimilarities*

  The assessment process or rather the support of this process by showing similar situations is such that the engineer looks at the already assessed situations and estimates a relative change of exposure based on the differences. However, this becomes more difficult the more "differences" exist. Therefore, the experts consider similar situations that exceed a certain number of differences as not helpful and want them removed.

Note that one can think of many other filters and parameters. Our goal (and contribution) is not to provide concrete parameters and filters, but to show general concepts and techniques for future extensions. The two filters mentioned are therefore practically relevant filters on the one hand and enable us to discuss (more technically) filter possibilities on the other hand. Before discussing these "personalization" options, we want to present the fundamental idea of displaying similar situations as supportive action in more detail first.

The first step is of course to calculate the similarity value of the "new" situation with the set of situations in the knowledge base. An example visualization is shown in Figure 3.46. The graph in this example uses a spring layout. The layout uses as "spring forces" the similarity values calculated by our similarity function. The example shows that this representation looks nice/interesting, but is obviously not very helpful for the assessment task.

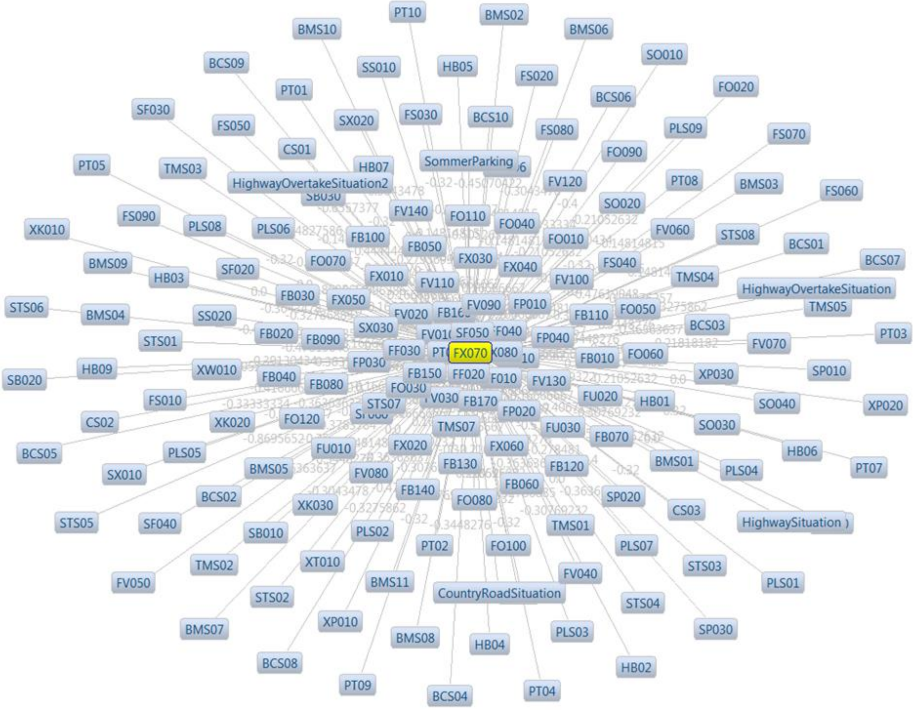


Figure 3.46: CBR "cloud" of similar situations

We therefore need a more sophisticated approach for presenting or dealing with similar situations. As discussed above, the fundamental idea is to present similar situations, reason about the change from one situation to another, and derive a corresponding change of the exposure value. In 1943 Lewin designed the so-called "Force Field Analysis" (FFA) [Lew43]. The FFA was designed to support decision-making in social psychology. Coming from social science, the technique aims at reaching a goal in terms of the change of a social situation. Due to the complexity of social situations, there was a need to support the decision process by structuring influential *forces*.

On the one hand, there are helping forces, figuratively pulling one towards the goal, and hindering forces blocking the movement towards a goal.

The transfer of FFA to the decision process of assessing the exposure of an HRA situation is (due to its simplicity) not too difficult. The desired state is, of course, the exposure value of the new situation, while the current state is given by one (or more) previously assessed situation(s). The forces are a little bit trickier. The goal is to derive an exposure value. Thus we interpreted the forces as pulling towards a higher exposure value or pushing towards a lower exposure value. The factors that are "pulling" or "pushing" are the influence factors from the dissimilarity sets (cf. $disSit$ in 3.4.2). The dissimilarity sets were (roughly) the influence factors contained in one situation, but not in the other situation. Let us assume without loss of generality that we have two situations Sit_{old} and Sit_{new} and the respective dissimilarity sets $disSit_{old}$ and $disSit_{new}$. Furthermore, we know that Sit_{old} has an exposure value assigned. All influence factors in the set $disSit_{old}$ are excrescent factors in the sense that they are not contained in Sit_{new}. If we were to remove all of these influence factors from the situation, we would make the situation more general and therefore the exposure would increase. Thus, the set $disSit_{old}$ is pulling towards a higher exposure. Analogously, the influence factors in the set $disSit_{new}$ make the situation more "special", resulting in a lower exposure. Thus, the set $disSit_{new}$ is pushing towards a lower exposure. Figure 3.47 shows a schematic example of the "forces" described above.

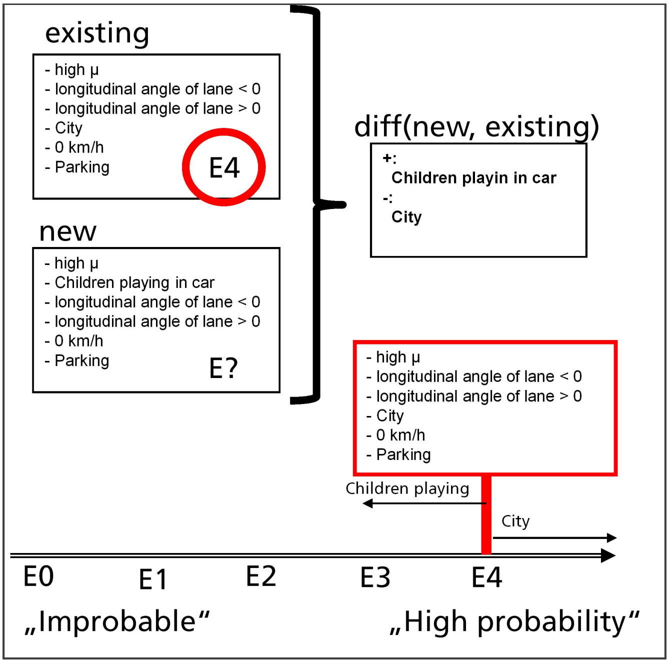

existing
- high µ
- longitudinal angle of lane < 0
- longitudinal angle of lane > 0
- City
- 0 km/h
- Parking
E4
new
- high µ
- Children playing in car
- longitudinal angle of lane < 0
- longitudinal angle of lane > 0
- 0 km/h
- Parking
E?
diff(new, existing)
+:
Children playin in car
-:
City
- high µ
- longitudinal angle of lane < 0
- longitudinal angle of lane > 0
- City
- 0 km/h
- Parking
Children playing
City
E0
E1
E2
E3
E4
„Improbable“
„High probability“

Figure 3.47: Example of applying FFA to the exposure decision based on situation similarity

In this example, the dissimilarities are $disSit_{old} = \{\texttt{City}\}$ and $disSit_{new} = \{\texttt{Childen playing in car}\}$. Additionally, the exposure value of the old situation is given as E4. Starting from E4 on the lower right, the forces are displayed towards a lower/higher probability. Note that the influence factors in the red rectangle are the common influence factors.

Figure 3.47 shows the forces of only one situation. We could extend the figure by adding more abstract (common) situations as red rectangles and drawing the forces based on the $disSit$ sets. We omit the display of such an example and assume that it is quite straightforward to understand that such a picture with more than three of these rectangles gets quite complex visually. Furthermore, while Figure 3.47 nicely shows the idea of FFA applied to exposure assessments based on situation similarity, the visual representation is not the key benefit. The purpose of Figure 3.47 is only to visualize the application of FFA to the HRA situation exposure assessment.

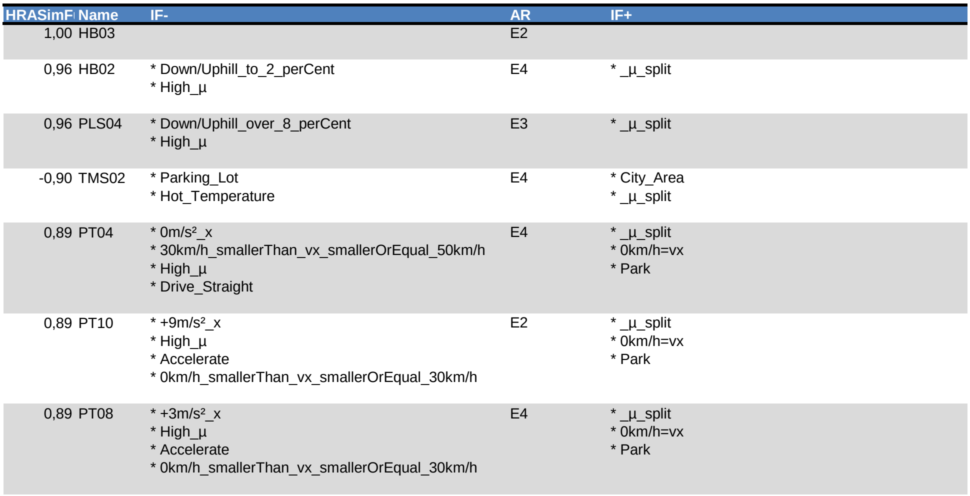
For the practical application we kept the FFA idea but used a more compact representation, a table. We explain our practical solution by using a small example "situation". Let us assume:

- City Area
- µ-split
- 0km/h=vx
- Park

Based on this information, we can query the knowledge base for similar situations. The result (produced by our approach and tool) is shown in Figure 3.48. Shown is a screenshot of an Excel table with five columns. The meaning of each column is as follows:

1. The column "HRASimFunc" shows the value calculated by the function $HRASimFunc$ (cf. (3.26)) introduced above.
2. The column "Name" is the name of the situation retrieved from the knowledge base, i.e., an existing situation the new situation is evaluated against.
3. The column "IF-" shows the set of influence factors the old situation has in addition to the new situation. This is the set $disSim_{old}$.
4. The column "AR" shows the assessment result in terms of exposure value.
5. The column "IF+" shows the influence factors the new situation has in addition to the old situation. This is the set $disSim_{new}$.

The example shows in the first row the case that a situation is already contained in the knowledge base (HRASimFunc = 1,00). Thus, we could short-cut the assessment process and just stick to the former assessment of being "E2". However, for explanatory reasons we assume the first (solution) row does not exist (i.e., HB03 is not contained in the knowledge base). The next row shows the situation with the name/id "HB02". This situation



HRASimF	Name	IF-	AR	IF+
1,00	HB03		E2	
0,96	HB02	* Down/Uphill_to_2_perCent * High_µ	E4	* _µ_split
0,96	PLS04	* Down/Uphill_over_8_perCent * High_µ	E3	* _µ_split
-0,90	TMS02	* Parking_Lot * Hot_Temperature	E4	* City_Area * _µ_split
0,89	PT04	* 0m/s²_x * 30km/h_smallerThan_vx_smallerOrEqual_50km/h * High_µ * Drive_Straight	E4	* _µ_split * 0km/h=vx * Park
0,89	PT10	* +9m/s²_x * High_µ * Accelerate * 0km/h_smallerThan_vx_smallerOrEqual_30km/h	E2	* _µ_split * 0km/h=vx * Park
0,89	PT08	* +3m/s²_x * High_µ * Accelerate * 0km/h_smallerThan_vx_smallerOrEqual_30km/h	E4	* _µ_split * 0km/h=vx * Park

Figure 3.48: Example FFA table

additionally states that the vehicle is parked on an in/declined plane and that there is "High μ" (cf. column *"IF-"*). The influence factor missing from the situation to be assessed, is "μ-split". (Note, that we would usually say it was exchanged with "High μ".) As explained above, the column AR shows, the assessment result; in this case "E4" (for HB02). The task would be to reason about the impact on the exposure when changing from "μ-split" to "high μ" and the additional (up to 2%) in-/declined plane.

Thus, by leveraging the HRASimFunc-Function and the transferred idea of Lewin's Force-Field-Analysis we have provided significant support for the decision process for HRA situation exposure.

As mentioned above, we want to present two practically important extensions of the support approach. The focus is not on the concrete filtering approach, but to show different (technical) extension approaches. We will present the following two "filters":

- Set filtering
- Restricted number of dissimilarities

A straightforward solution to all of these personalization filters is to define rules and filter the similarities after calculation. Given the current size of the knowledge base, this is not a problem, but it is not efficient on the one hand and not necessary on the other hand. What we want is to restrict the case-based reasoning, thus, the calculation of HRASimFunc values upfront. The "cases" stem from our ontology; thus, we need to first "tell" the ontology which situations are relevant and use only those as a basis for case-based reasoning. The disadvantage, however, is that one might want to filter the results *after* they are presented (based on what he sees). We will first present ontology changes for the pre-reasoning-filters and afterwards explain our solution for post-reasoning filters.

The first filtering option was to define sets of wanted and unwanted influence factors. The first part, the influence factors we want to have, is easy to determine using the ontology. We simply define a class that contains all situations "consistingOf" the set of influence factors. This process is the same we used for classifying the situations into a hierarchy using the ontology reasoner (cf. section above). The nice "feature" we did not mention is that not only the classes are classified, but the instances of the situations are also assigned automatically assigned to the respective classes. This means that we simply define the class as mentioned, run the reasoner/classifier, and feed the situation instances of the result into the case-based reasoning process.
The opposite case, the negative list, is more difficult. Due to the Open World Assumption underlying OWL, we cannot just define that something is "not" the case. If we describe a situation by the set of influence factors *highway, driving_straight* does, for example, not mean that it is not raining!
In an open world scenario we need something we can refer to with "not". Thus, we need to introduce closure into the ontology (at least for the situation example). One possibility is to define a class not by describing properties that must be true for an instance to be a member of the class, but explicitly as a set of (disjoint) instances.

The solution process is therefore:

1. Use the class value restriction to define positive and negative lists of situations.
2. Take these sets of situations to make the situations disjoint.
3. Create for each set (positive and negative set of influence factors) a class defined by the respective set.
4. Define a new class, which is equivalent to the class describing the set of situations compliant with the positive list criteria and not with the class describing the set of situations compliant with the negative list.

The process is visualized/exemplified in Figure 3.49.

Finally, filtering by the number of dissimilarities is only possible once we know what the dissimilarities are. These are determined during case-retrieval. Thus, we cannot restrict this through ontological definitions upfront. Once we have retrieved the data for displaying the table, we can simply count the elements in the columns "IF-" and "IF+" and omit the situation if it does not fulfill the maximal dissimilarity restriction.

We conclude this section by summarizing that we presented an ontology-based exposure assessment support (with practicable extensions) on the one hand, and provided a technique for checking the consistency of the knowledge base on the other hand.

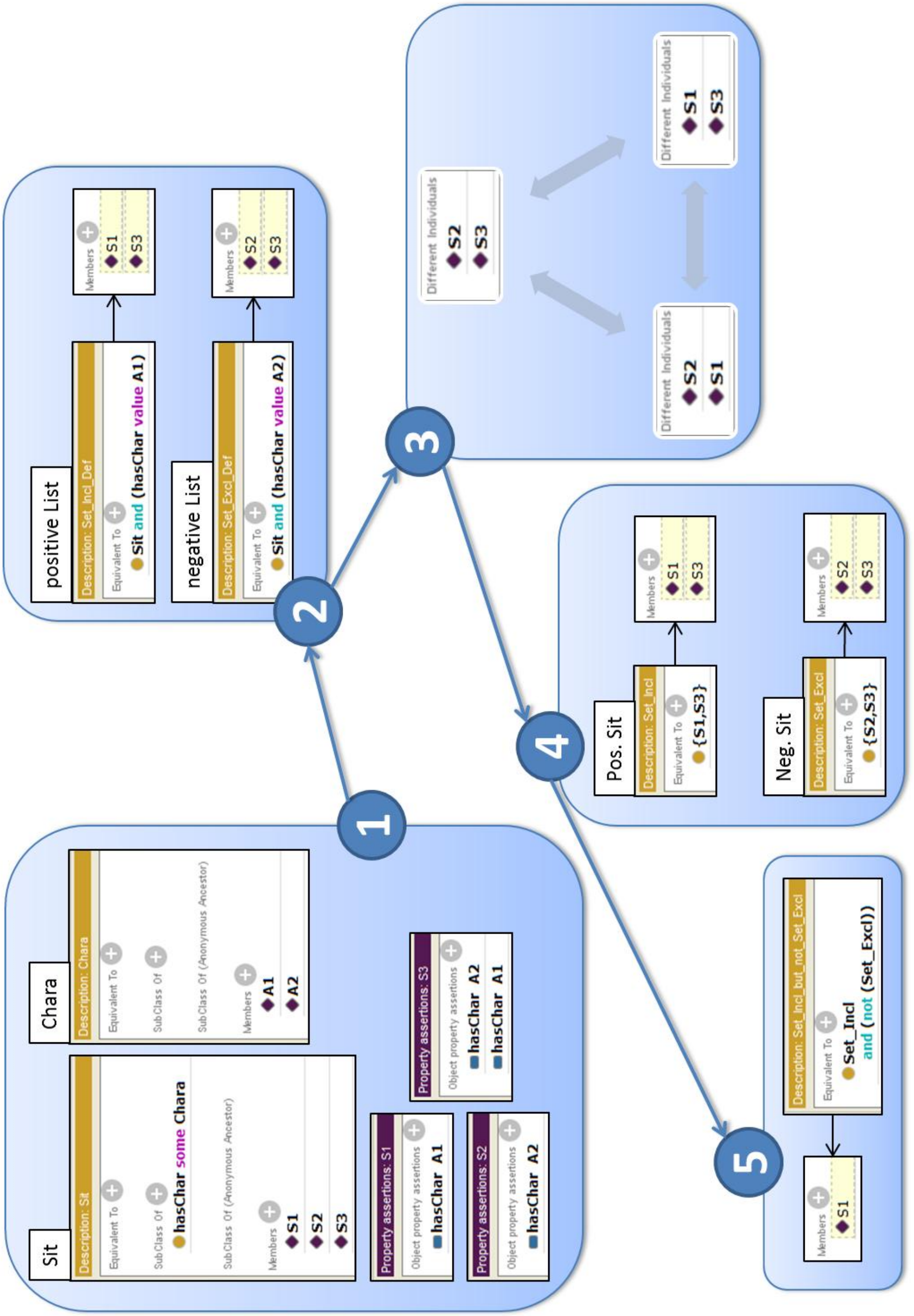


Figure 3.49: Ontology-based situation filtering incl. negative lists

3.5 ARID - Analysis of Risk through In-system Degradation

In the previous sections, we introduced with GOBI a formalization and integration framework. We furthermore discussed the formalization of situation influences and situation assessments in the previous two sections, OASIS and HEAT. The other fundamentally important factor for HRAs, besides operational situations, is the hazard itself. Figure 3.50 shows the integration of HEAT with the overall thesis on the one hand and the general structure of this section on the other hand.

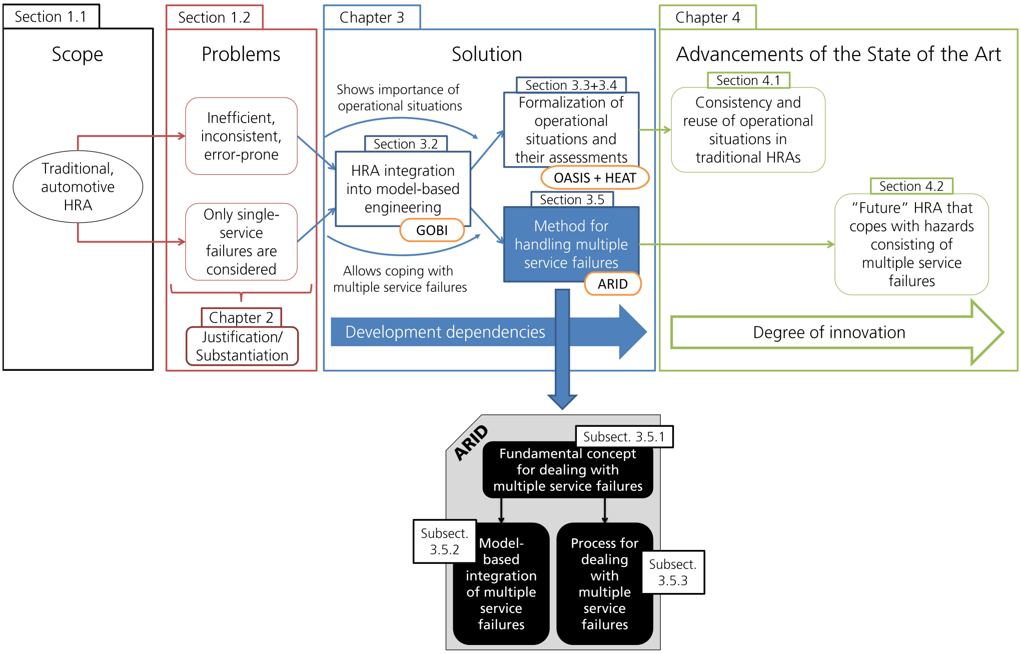

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method for handling multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation
ARID
Subsect. 3.5.1
Fundamental concept for dealing with multiple service failures
Subsect. 3.5.2
Model-based integration of multiple service failures
Process for dealing with multiple service failures
Subsect. 3.5.3

Figure 3.50: Integration overview of ARID

The main focus of this section is therefore on risk assessment. In section "2.2 Core Challenges and Goals" the breakdown of fundamental challenges was discussed:

✧ Structuring of risk assessments
  - ✓ Hazardous event classification
  - ➠ Dealing with multiple service failures

The challenge of risk assessments is the hazardous event classification, i.e., performing an assessment and handling the enormous number of hazards if not only single service failures are considered, but multiple service failures as well.
This classification inherits the challenges of formalizing and structuring con-

trollability and severity estimations. However, these challenges are not the subject of this thesis. For ARID, we take the traditional approach, as performed in practice, of dealing with hazardous event classification. Our focus in this section is on the challenge of dealing with multiple service failures.

If we are talking about a safe vehicle (in the sense of functional safety), we assume that *all* risks are acceptable. Hence, we imply or assume that there is no event that results in a harm scenario that is not acceptable.

Technically, however, acceptability of *all* risks implies that we cannot stop with assessing single feature failures, but also need to consider hazards consisting of more than one failing feature. This statement is true in its genericity, but becomes obvious if we consider two examples:

1. Assume we have an active steering system and a power train system (with the respective features). If we look at the system from an external feature-oriented perspective, we would assess the risks of both features independently. If we assume that (due to the trend towards high integration of functions on one ECU) both systems are sharing the same ECU, the question arises of what the risk is if the ECU is failing in such a way that both features are failing dangerously. In this example, one would obviously expect that this situation, i.e., a hazard comprising two feature failures, is considered in the risk management process. If we look at the immanent trend towards ever increasing integration of functions on shared platforms, the question arises of how critical a common cause (of multiple features) is. Is the assumption valid that a platform inherits the maximal ASIL of the functions residing on it (or better: their requirements)? If we consider one platform containing an ASIL C function and compare this to a platform containing 10 ASIL C functions, is it valid that both platforms are developed with the same rigorousness?
2. Another scenario is given in the context of system-level redundancy. By system-level redundancy we mean any other feature or service that can compensate (at least to a certain extent) the failing of a service. In automotive systems, system-level redundancy is reflected by the "controllability"-assessment. "Controllability" allows reducing the risk emerging from one system because the driver (or any other person) can substitute the expected functionality by leveraging another function. A concrete controllability argument is that a "self-acceleration" failure is not too critical because the driver can "over"-brake the self-acceleration. This means: If the car is accelerating without the driver's request, the driver can still stop the car by using the brakes (assuming the brakes are stronger than the engine, which is enforced by law). Thus, the risk emerging from the power-train system is reduced by relying on the functioning of the brakes. If both systems are additionally subject to internal dependencies as discussed in the first example, the case is obvious; we need an additional risk assessment. But even if we assume that both systems are totally independent (in their realization), the question remains whether the risk of both systems failing at the same time is acceptable. Shouldn't

there be a (safety) requirement for ensuring independence in the realization with a rigorousness derived from the risk that both systems are failing?

The two cases introduced above are both examples in which the features are dependent. In the first case, we had an internal dependency situation; in the second case, an external dependency situation.

Independent of a concrete situation, the challenge remains that in a worst-case scenario, we have to assess the power set of feature failures. The following section is therefore structured into three sub-sections:

1. ARID Fundamentals
   In this section, we present the modeling basics and fundamental considerations for handling multiple service failures in an HRA.
2. ARID Concepts
   In this section, we explain which knowledge we reuse and how we can leverage this knowledge in the process of further hazard assessments.
3. ARID Analysis Process
   In this section, we put the fundamentals and the concepts together and describe a concrete analysis process.

While the first two sub-sections are theoretical, the last part describes the actual analysis. The theoretical sub-sections discuss the general idea and the mathematical background of ARID, we will discuss the usage of ARID and optimizations of the general theory in the analysis sub-section. The optimization approach relies on OASIS and HEAT. We will therefore explain their integration and interaction with ARID as part of the optimization discussion.

After reading this section, the reader will know, on the one hand, how models and their formal knowledge can be used to deal with multiple service failures and, on the other hand, how a model-based approach for HRAs, such as SAHARA (with OASIS 3.3 and HEAT cf. 3.4), enables us to deal with multiple service failures.

3.5.1 ARID Fundamentals

In order to understand the basic idea of ARID, we will summarize the tasks and goals of safety engineering. The fundamental goal is to end up in a system that only exposes acceptable risk(s). In order to achieve this, the safety engineer essentially has to perform two tasks: the identification and assessment of risks, which results in necessary risk reductions, and the implementation or realization of the risk reductions. Traditionally, these two steps have been loosely coupled or only connected via the "safety goals". This is even known figuratively as the bow tie diagram of hazard analysis (cf. also 2.7). Regarding this aspect, one can see a huge gap between industrial practice and scientific advancement. From a practical point of view, we see the biggest improvement in hazard analysis in the clean separation of hazards from system development. However, we only claim this because hazards and causes for hazards are often mixed deliberately (in industrial practice). If we assume that the separation of the terms and concepts is clear (and fulfilled), as it is in the scientific community, a huge step forward can be made by integrating the hazard analysis into the system development (without losing the distinction between a hazard and its causes). We therefore propose integrating the realization of safety goals and the derivation of safety goals more closely.
We propose this because especially with the challenge of exponential number of hazards ahead, we need to limit ourselves to risk assessments that are relevant! The relevance of an assessment can, however, only be determined in the realization/implementation part: the system development. Thus, a risk assessment is only relevant iff its result has an impact on the development of the system.

For the systematic reduction of risk assessments, the negation is maybe even more interesting: We do not need to perform a risk assessment if we know that the result will not have an impact on the development of the system!

The fundamental question, however, is how we can know that an assessment does not impact the further development. We find the answer in the realization phase of the safety goals. Safety goals are broken down to components and units. Whether a component is relevant is given by error propagation models. If there is a cause that could lead to a violation of the safety goal, the cause is assigned a safety requirement that ensures that the cause does not appear (with a required likelihood). With this very coarse summary of the process of safety engineering, we can explain the relevance of a risk assessment. If we assume that a cause has "received" a certain safety requirement ensuring that the cause does not occur, we can state that a risk assessment that results in the same (or a less strict) safety requirement will not yield new information or new restrictions and is therefore not necessary.

In order to work with safety requirements in a more formal way, we reduce safety requirements to a maximal failure rate (or later a maximal occurrence probability). Note that this is compliant with most safety standards. As discussed in the state-of-the-art section, most safety standards define some

kind of safety integrity level. ISO 26262 defines Automotive Safety Integrity Levels. These levels can be translated into failure rates that need to be ensured. For automotive systems, the maximal integrity level is ASIL D, which can be translated into a failure rate of less that 10^{-8}.
We explain the idea of ARID informally discussed above using an abstract example.

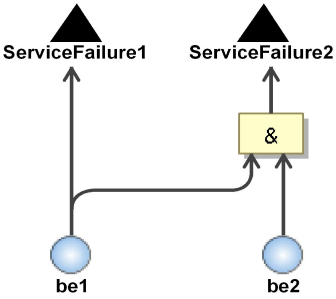


Figure 3.51: Tiny example showing two fault trees for exemplifying the ARID idea

Figure 3.51 shows a fault tree consisting of two top events: ServiceFailure1 and ServiceFailure2. Additionally, two basic events are shown (be1 and be2). The (very) basic idea of ARID exemplified in this figure is that if we assume we are in an automotive context, the maximal safety requirement (with ASIL D) is quantitatively the requirement of being smaller than 10^{-8}, as introduced above. Let us assume that one hazard is associated with ServiceFailure1 and the risk assessment result is that ServiceFailure1 must not have a failure rate higher than 10^{-8}; thus it has an ASIL D safety requirement. But in this we have not used our knowledge to the full extent. If ServiceFailure1 needs to have a failure rate no higher than 10^{-8} and "be1" is the only cause for this service failure, we can derive that the basic event "be1" must not have a higher failure rate than 10^{-8}. With this knowledge, however, we can derive that the combined failure rate of "$be1 \wedge be2$" cannot be higher than 10^{-8} (independent of the concrete failure rate of "be2"!). Due to the fact that ServiceFailure2 consists exactly of this combination, we can derive that due to the assessment of (the risk associated with) ServiceFailure1, ServiceFailure2 cannot have a higher failure rate than 10^{-8}.
Thus, if we were to assess the hazard consisting of ServiceFailure2, the maximal requirement we could get is 10^{-8}, but this is already ensured. Hence, there is no need to manually assess this hazard! (We would not get more safety requirements than we already have.) However, we should retain the information that the hazard has the requirement of 10^{-8}. Note that this is a very simplified, synthetic example for showing the basic idea of ARID. However, it shows that by storing the risk assessment result in the system-

internal model as a development constraint, we can reuse that knowledge to reduce the number of hazards to be analyzed.

Model Basics

As described above, ARID strongly relies on knowledge (contained in models) about:

- the system (with respect to features and their dependencies)
- the failure propagation and, especially, the basic event (cause) distribution
- the assessment result in terms of quantitative development constraints (ASILs)

As introduced with GOBI, we already have a formalization for important terms and concepts such as feature or service. This formalization is important for ARID as well. However, it represents only an abstract mathematical formalization and not a concrete model-based approach. We will not provide a new model-based approach, but will use common, practically relevant modeling approaches for the concrete discussion of ARID.
Please note that our approach does not rely on the specific models presented. It can be transferred to other model types if they are able to express the required concepts.

In the following, we will present the required concepts for each of the above-stated three models.

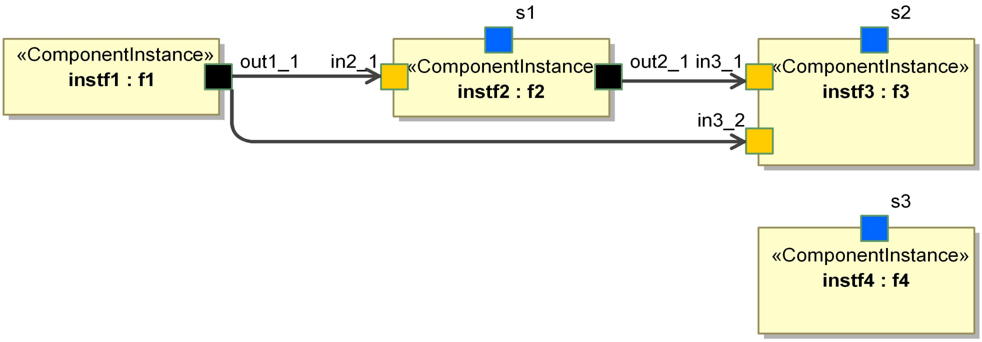


Figure 3.52: Introductory functional network with service annotation

Figure 3.52 shows an example of a service-annotated functional network. Shown are four functions ("instf1:f1", "instf2:f2", "instf3:f3", "instf4:f4") with three of the functions being connected in a network and the fourth function "instf4:f4" not connected. The connected functions are connected via inports and outports, which reside on the sides of the functions: Inports are on the left side; outports on the right side. Additionally, there are blue ports displayed on the top of the three rightmost functions. These ports constitute the service the function delivers.

More formally, we can define a function f with inports P_I, outports P_O, and services S as:

$$f = (P_I, P_O, S) \tag{3.35}$$

We use the capital letter version to refer to a set of functions and add a tuple label if appropriate:

$$\begin{aligned} F_{\mathcal{A}} = \{\text{''}function\ name\ 1\text{''} &= (\, P_{I_1}, P_{O_1}, S_1\ ) && (3.36)\\ \text{''}function\ name\ 2\text{''} &= (\, P_{I_2}, P_{O_2}, S_2\ ) && (3.37)\\ &\vdots \\ \text{''}function\ name\ n\text{''} &= (\, P_{I_n}, P_{O_n}, S_n\ )\} && (3.38) \end{aligned}$$

A connection relation from outports to inports is defined by:

$$\begin{aligned} R_{connect} \subseteq P_O \times P_I \text{ with } (p_o, p_i) \in R_{connect} \\ \text{iff outport } p_o \text{ is connected to inport } p_i \end{aligned} \tag{3.39}$$

Formally expressed, the example from above is therefore:

$$\begin{aligned} F_{\mathcal{A}} = \{\text{''}instf1 : f1\text{''} &= (\ \{\}, \{out1\_1\}, \{\}\ ), && (3.40)\\ \text{''}instf2 : f2\text{''} &= (\ \{in2\_1\}, \{out2\_1\}, \{s1\}\ ), && (3.41)\\ \text{''}instf3 : f3\text{''} &= (\ \{in3\_1, in3\_2\}, \{\}, \{s2\}\ ), && (3.42)\\ \text{''}instf4 : f4\text{''} &= (\ \{\}, \{\}, \{s3\}\ )\ \} && (3.43) \end{aligned}$$

$$R_{connect} = \{(out1\_1, in2\_1), (out1\_1, in3\_2), (out2\_1, in3\_1)\} \tag{3.44}$$

Besides this service-annotated functional network, we need system-internal knowledge about the causes why services fail. There are many approaches available to model cause-effect relations. We refer to Domis [Dom12] for an extensive discussion of failure logic models and also use the approach he uses: Component Integrated Component Fault Trees [ADH$^+$10]. Figure 3.53 shows a failure logic model as a Component Fault Tree.

The essential parts of such a failure logic model are:

- A set of service failures SF
- A set of basic events BE
- A Boolean logic formula connecting service failures and basic events; any representation of a Boolean logic formula can be used. Without loss of generality we use the canonical sum of products representation:

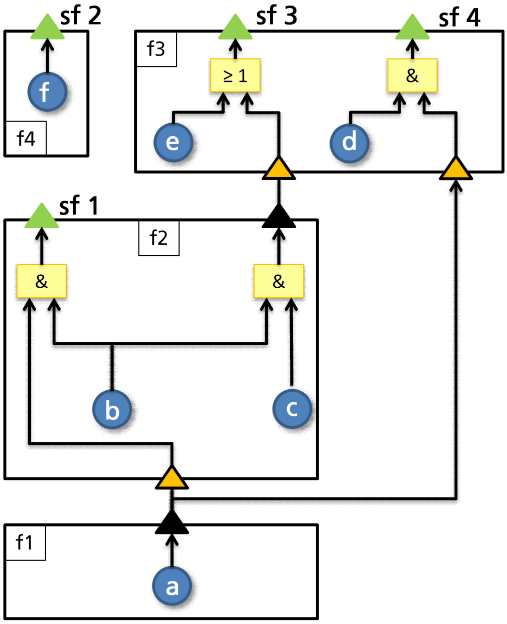


Figure 3.53: Introductory failure logic model as Component Fault Tree

$$f_{FT}\left(sf\right) = \vee_i \wedge_j \left(be_{ij} \vee \overline{be_{ij}}\right) \text{ with } \; be_{ij} \in BE \tag{3.45}$$

Formalized with the introduced definitions, the example is:

$$SF = \{sf1, sf2, sf3, sf4\} \tag{3.46}$$

$$BE = \{a, b, c, d, e, f\} \tag{3.47}$$

$$f_{FT}\left(sf1\right) = a \wedge b \tag{3.48}$$

$$f_{FT}\left(sf2\right) = f \tag{3.49}$$

$$f_{FT}\left(sf3\right) = b \wedge c \vee e \tag{3.50}$$

$$f_{FT}\left(sf4\right) = a \wedge d \tag{3.51}$$

Note that Figure 3.53 additionally shows which basic event belongs to which function (cf. the black boxes in Figure 3.53). This can be expressed by the following formal relation:

$$R_{FunctionFaults} : F \times BE \text{ with} \tag{3.52}$$

$$R_{FunctionFaults} = \{(f, be) \in F \times BE| \tag{3.53}$$

$$\text{Basic event } be \text{ is contained in function } f \; \} \tag{3.54}$$

In our example $R_{FunctionFaults}$ holds the following tuples:

$$\begin{aligned} R_{FunctionFaults} = \{ & (f1, a), && (3.55) \\ & (f2, b), (f2, c), && (3.56) \\ & (f3, d), (f3, e), && (3.57) \\ & (f4, f) \} && (3.58) \end{aligned}$$

Now we are almost set to present our approach for dealing with hazards consisting of multiple service failures. However, we need to define hazards more formally. For ARID, a hazard is a set of service failures:

$$H = P(SF) \tag{3.59}$$

We further define an abbreviated notation for the cardinality of a hazard:

For h with $|h| = n$ we write: h_n (3.60)

Analogously we denote with H_n the set of all hazards consisting of n service failures. Hence, we can constructively define the set of hazards of a system with the set of service failures SF as:

$$H = \bigcup_{n=1}^{|SF|} H_n \tag{3.61}$$

Note that by definition cf. 3.59, the Boolean logic formula describing the failure logic for a hazard to occur is:

$$F_H(h) = \bigwedge_{sf \in h} f_{FT}(sf) \tag{3.62}$$

ARID's basic idea

In this section, we use the above-discussed ARID fundamentals and introduce the basic idea of ARID on a conceptual level.

The first aspect we will discuss in more detail is the formal background of quantitative safety requirements. The quantitative part of a safety requirement is usually given as the maximally allowed failure rate (or hazard rate). It is often given as a failure in time (FIT) value and is denoted by the Greek letter λ. One FIT is defined as one failure in 10^9 hours. This definition does, however, imply a constant failure rate. In general, the failure rate is time dependent $\lambda(t)$. But if a safety requirement states that a certain failure rate should not be exceeded, it means that this requirement is a kind of constant boundary. The actual technical failure rate can be time-dependent, but one must ensure that the failure rate does not exceed the given boundary at any time. Our approach deals with eliciting the requirements, and we can therefore always assume a constant failure rate.

Often (mistakenly) unified is the occurrence probability (of a failure). However, a probability always requires the definition of a certain period of time. If we assume a continuous process, the failure probability is defined by:

$$P(T \leq t) = F(t) = \int_0^t f(\tau)\, d\tau \tag{3.63}$$

In this equation, $F(t)$ is the failure distribution function and $f(t)$ is the failure density function. If we assume a constant failure rate (as discussed above), the respective density function is the exponential density function $f_e(t)$.

$$f_e(t) = \lambda e^{-\lambda t} \tag{3.64}$$

The corresponding failure distribution (and probability function), termed exponential failure distribution $(F_e(t))$, is:

$$P(t) = F_e(t) = \int_0^t \lambda e^{-\lambda \tau} d\tau = 1 - e^{-\lambda t} \tag{3.65}$$

For a given mission time t_m and exponential distribution, we can therefore transform the failure rate into a probability and vice versa. According to the German Kraftfahrt-Bundesamt (Federal Motor Vehicle Office), the average age of vehicles in Germany (as of 01.01.2013) is 8.7 years [KB13]. If we take this as the mission time of a vehicle we can transform the failure rate requirements into probability requirements:

We know, of course, that the commonly accepted safety engineering practice is to use failure rates as the fundamental model. Due to the fact that calculations with failures rates are very difficult, many tools, such as Fault

Table 3.4: Failure probability based on constant failure rate and 8.7 years of mission time

λ	$P(8.7a)$
10^{-04}	100%
10^{-05}	53%
10^{-06}	7.3%
10^{-07}	0.76%
10^{-08}	0.076%
10^{-09}	0.0076%

Tree tools, use the duality of failure rates and probabilities and do internal calculations with probabilities. We subscribe the same idea and therefore use probabilities in our approach. In the following, we will discuss the development implications of a risk assessment.

Risk assessment results in a quantitative safety goal to be achieved. The responsibility of the subsequent safety- and systems engineering is to ensure that this goal is achieved. In order to fulfill this, one reduces the occurrence likelihood of causes of the unwanted top-event. In fault trees, causes and their relations are given by basic events. The logical formula additionally introduces the knowledge about which basic events must occur together to trigger the top-event. A canonical representation of these necessary and sufficient sets of basic events (derived from Boolean logic formulas) is the prime implicant (or minimal cut set) representation.

Given a Boolean logic formula f (such as our f_{FT} defined above), an implicant of f is a product term p that implies f.

$$\begin{aligned}\left(p \textit{ is implicant of } f\right) &\Leftrightarrow \left(p \Rightarrow f\right)\\ &\Leftrightarrow \left(\forall \overrightarrow{x} p\left(\overrightarrow{x}\right) = 1 \Rightarrow f\left(\overrightarrow{x}\right) = 1\right)\end{aligned} \tag{3.66}$$

We can furthermore define that a *prime* implicant of a function is an implicant that is minimal. The interesting aspect of prime implicants is that a (Boolean logic) function can be represented by a complete set of prime implicants. This representation is called Blake canonical form [Sas96]. It is a disjunctive normal form (that is not necessarily minimal) [Bro03]. We assume that $\mathcal{P}\left(f_{FT}\right)$ extracts the set of prime implicants of a Boolean logic (fault tree) formula. We use $\wp \in \mathcal{P}\left(f_{FT}\right)$ to refer to a prime implicant of f_{FT}.

Let us assume that we have a service failure sf and its associated failure propagation model $f_{FT}\left(sf\right)$, as defined above. If we assess the risk of the hazard consisting of this service failure, we end up with some requirements for avoiding the service failure to occur. Quantitatively, we assign to the top-event a safety requirement with a maximal failure rate of $\leq 10^{-\alpha}$: $\lambda_{req}\left(f_{\mathsf{FT}}\left(sf\right)\right) \leq 10^{-\alpha}$.

We furthermore know that based on the fact that the requirement is a constant failure rate (as a boundary) and that we can assume an average mission time of 8.7 years, we can translate the failure rate into a maximal occurrence probability over the lifetime:
$P_{req}\left(f_{\mathsf{FT}}\left(sf\right)\right) \leq 10^{-\beta}$. It holds that:

$$P_{req}\left(f_{\mathsf{FT}}\left(sf\right)\right) = 1 - e^{\lambda_{req}\left(f_{\mathsf{FT}}\left(sf\right)\right)*8.7a} \tag{3.67}$$

$$10^{-\beta} = 1 - e^{10^{-\alpha}*8.7a} \tag{3.68}$$

or

$$\beta = -\frac{\ln\left(1 - e^{10^{-\alpha}*8.7a}\right)}{\ln 10} \tag{3.69}$$

In the following, we will use probability as a requirement. Due to the characteristics of being a prime implicant, we can derive the following necessary requirement for the prime implicants of $f_{\mathsf{FT}}\left(sf\right)$:

$$\forall \wp \in \mathcal{P}\left(f_{\mathsf{FT}}\left(sf\right)\right) : P_{req}\left(\wp\right) \leq 10^{-\beta} \tag{3.70}$$

The development constraint is that the actual probability of a prime implicant $P_{act}\left(\wp\right)$ to occur must be in the interval: $\left[0, 10^{-\beta}\right]$.

If we store these prime implicant requirements and assume that they will be fulfilled during system development, we can systematically reduce the number of hazards to be assesses. We will explain this using an abstract example.

We assume that we have already assessed some hazards and have a corresponding set of prime implicants $\mathcal{P}^{\surd}$ with quantitative requirements assigned. If we now consider a new hazard h and an implied failure propagation formula $f_{FT}\left(h\right)$, we can calculate the set of prime implicants of this function: $\mathcal{P}\left(f_{FT}\left(h\right)\right)$.
Given these two sets of prime implicants, it might be that we already have requirement boundaries for (some of) the prime implicants in $\mathcal{P}\left(f_{FT}\left(h\right)\right)$. We obviously have concrete boundaries for the set: $\mathcal{P}\left(f_{FT}\left(h\right)\right) \cap \mathcal{P}^{\surd}$. But we can furthermore derive boundaries based on the information of the given requirements. This reasoning is based on the fact that implicants can be seen as sets of literals (in one conjunction). The usual construct of a subset relation induces a partial order into the set of literals. This means that we can order each set of literals (and by this the implicants) partially. If the set of literals is, for example, $\{a, b, c\}$, we can build a partial order as displayed in Figure 3.54.

Each element in the partial order is a conjunction of literals. Each element logically implies its descendents (bigger elements). In Figure 3.54, the ar-

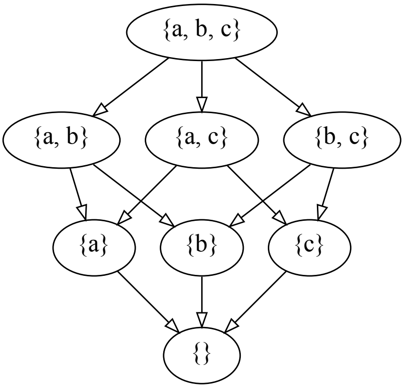


Figure 3.54: Partial order of implicants

rows can therefore be interpreted as logical implications (if the nodes are conjunctions of the contained literals). For example: $a \wedge b \Rightarrow a$

The implication can be used for the derivation of the probability requirements as well. If we assume that there are two independent events a and b, and we know that $P_{req}(b) \leq \varsigma$, we know that:

$$\text{if } a \Rightarrow b \text{ and } P(b) \leq \varsigma \text{ then } P(a) \leq \varsigma \tag{3.71}$$

Transferred to our prime implicants example, we derive that if one prime implicant $\wp^?$ of the new hazard ($\wp^? \in \mathcal{P}\left(f_{FT}\left(h\right)\right)$) implies one prime implicant for which we have a requirements boundary, it inherits that requirement.

$$P_{act}(\wp^?) = \min\left\{P_{act}(\wp) \mid \wp^? \Rightarrow \wp \wedge P_{act}(\wp) := P_{req}(\wp)\right\}, \forall \wp \in \mathcal{P}^{\surd} \tag{3.72}$$

We can now furthermore derive that the hazard to be assessed (or the respective top event) has at least the probability boundary of the prime implicant with the *maximally valued* requirement boundary (because the function is a *disjunction* of prime implicants).

$$P_{act}\left(f_{FT}\left(h\right)\right) = \max\left\{P_{act}(\wp)\right\}, \forall \wp \in \mathcal{P}\left(f_{FT}\left(h\right)\right) \tag{3.73}$$

In the introduction of this section, we stated that each domain has a minimal boundary of requirement. For avionics, this boundary is a failure rate of 10^{-9}; for automotive, we have a failure rate of 10^{-8}, which is equivalent to a probability over mission time (of 8.7 years) of 0.076% (cf. 3.4). Thus, if equation 3.73 results in the following actual constraint:

$$P_{act}\left(f_{FT}\left(h\right)\right) \leq 0.076\% \tag{3.74}$$

then there is no need to assess the risk of hazard h!

3.5.2 ARID Concepts

Knowledge reuse approach

The simple technique introduced above can be used already; however, it is very limited in its capability because we propagate only the minimal requirement to the prime implicants and the maximal requirement to the top event. In order to understand why this is not very sophisticated knowledge, we come back to our abstract partial order shown in Figure 3.54. If we assume that we want to assess a hazard that has $\{a, b\}$ as prime implicant and if we additionally know that $P(\{a\}) = P(\{b\}) = 0.5$, the algorithm introduced above can only derive that $P(\{a, b\}) \leq 0.5$. Due to the fact that all variables (literals) are independent, we know from logics and probability calculation that

$$P(\{a, b\}) = P(\{a\}) * P(\{b\}) = 0.5 * 0.5 = 0.25 \tag{3.75}$$

More generally speaking, the task is to determine for a given prime implicant a partition into disjoint implicants that delivers the minimal result. If, for example, we again consider Figure 3.54 and if we further assume that we want to know the actual requirements implication of the top-most implicant: $\{a, b, c\}$, then we could, of course, determine $P_{act}(\{a\})$, $P_{act}(\{b\})$, and $P_{act}(\{c\})$ and multiply the results as shown above. Due to the fact that a set of literals can have a stricter requirement than the multiplication of the single parts of the set, another possibility for the determination of $P_{act}(\{a, b, c\})$ could be to multiply:

$$P_{act}(\{a, b\}) * P_{act}(\{c\}) \text{ or} \tag{3.76}$$

$$P_{act}(\{a, c\}) * P_{act}(\{b\}) \text{ or} \tag{3.77}$$

$$\vdots$$

For only one prime implicant to decide which actual requirements obligation the implicant has, we need to find an optimal combination, which is obviously not trivial.

In order to formalize this we need some prerequisites:

- We need to formalize the **disjointness** of sets of literals (in order to be able to multiply the probabilities)
- We need to determine the set of all possible **combinations** of disjoint subsets ({a, b} + {c}, {a, c} + {b}, ...
- We need to calculate the **minimal actual requirement** of the sets of disjoint subsets.

Disjointness

Let $\wp_1$ and $\wp_2$ be two sets of literals (like prime implicants). We define $\widehat{\mathcal{P}}(\wp_1, \wp_2) : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$ as a function that calculates the commonly implied "implicants" over the set of literals $\mathfrak{L}$ by:

$$\widehat{\mathcal{P}}(\wp_1, \wp_2) = \left\{\wp | \wp \in 2^{\mathfrak{L}} \setminus \{\} \wedge \wp_1 \Rightarrow \wp \wedge \wp_2 \Rightarrow \wp\right\} \tag{3.78}$$

(Note that we used $2^{\mathfrak{L}}$ to refer to the power set of $\mathfrak{L}$.)
For example:

$$\widehat{\mathcal{P}}(\{a, b, c\}, \{b, c, d\}) = \{\{b\}, \{c\}, \{b, c\}\} \tag{3.79}$$

We can now define the disjointness decision function as:

$$\mathcal{X}_{disjoint}(\mathcal{P}) = \begin{cases} 1 & \text{if } \forall \wp_1, \wp_2 \in \mathcal{P} : \wp_1 \neq \wp_2 \Rightarrow \widehat{\mathcal{P}}(\wp_1, \wp_2) = \emptyset \\ 0 & \text{else} \end{cases} \tag{3.80}$$

Thus,

$$\text{Sets of sets of literals } \wp_1 \ldots \wp_n \text{ are disjoint} \tag{3.81}$$

$$\Leftrightarrow \mathcal{X}_{disjoint}(\{\wp_1, \ldots \wp_n\}) = 1 \tag{3.82}$$

Combinations of disjoint sets

The next step is to determine all combinations of disjoint sets of prime implicants. Let $\wp$ be the prime implicant we are analyzing and $\mathcal{P}^{\surd}$ the set of already assessed prime implicants:

$$\rho(\wp, \mathcal{P}^{\surd}) = \left\{\widetilde{\mathcal{P}} | \widetilde{\mathcal{P}} \subset \mathcal{P}^{\surd} \wedge \mathcal{X}_{disjoint}(\widetilde{\mathcal{P}}) \wedge \forall \wp' \in \widetilde{\mathcal{P}} : \wp \Rightarrow \wp'\right\} \tag{3.83}$$

Minimal Actual Requirement of $\wp$

We can now define our more sophisticated knowledge extraction function in the following way:
Let us again assume a set of already assessed prime implicants $\mathcal{P}^{\surd}$ with requirements boundaries and an unassessed set of prime implicants $\mathcal{P}(f_{FT}(h))$. For each $\wp^? \in \mathcal{P}(f_{FT}(h))$ we can derive:

$$P_{act}(\wp^?) = \min \left\{ P_{act}(\wp) \mid P_{act}(\wp) := \prod_{\wp \in \mathcal{P}^?} P_{req}(\wp) \right\}, \forall \mathcal{P}^? \in \rho(\wp^?, \mathcal{P}^{\surd}) \tag{3.84}$$

What is done in equation 3.84 is that we build all disjoint sets of prime implicants that are implied by $\wp^?$ and then calculate the probabilities of the elements in these sets. Finally, we derive the minimum as an already implied requirement for $\wp^?$.

Problem Complexity

While this equation works in theory, in practice the challenge is how to calculate this. In the worst case. the set of all disjoint sets has exponential size (exponential with the number of literals). The problem to solve is NP-hard. We prove this by showing that a maximum clique problem can be reduced to an instance of our prime-implicant minimization problem!

Let us assume that we have an undirected graph $G = (V, E)$. A clique in the graph is a subset of vertices $C \subseteq V$, such that for every two vertices in C, there exists an edge connecting the two. Let us formalize the edge decision:
Let v_1 and v_2 be vertices in V:

$$\mathcal{X}_{edge}(v_1, v_2) = \begin{cases} 1 & \text{if } \exists e \in E : e = \{v_1, v_2\} \\ 0 & \text{else} \end{cases} \tag{3.85}$$

We furthermore define a clique decision function for a subset of vertices $C \subseteq V$ and the graph G as:

$$\mathcal{X}_{clique}(C, G) = \begin{cases} 1 & \text{if } \forall v_1, v_2 \in C \text{ with } v_1 \neq v_2 \,|\, \mathcal{X}_{edge}(v_1, v_2) = 1 \\ 0 & \text{else} \end{cases} \tag{3.86}$$

Let $\overline{C} = \{C' \subseteq V | \mathcal{X}_{clique}(C', G) = 1\}$ be the set of all cliques of a graph G. A maximum clique C_{max} in G is:

$$C_{max} \in \overline{C} \wedge \forall C \in \overline{C} \,|\, |C| \leq |C_{max}| \tag{3.87}$$

The transformation into our prime-implicant minimization problem is given by the following.

1. We define the set of *missing* edges in G as $\overline{E}$:
   $\overline{E} = \{\overline{e} \,|\, e = \{v_1, v_2\} \wedge v_1, v_2 \in V \wedge \mathcal{X}_{edge}(v_1, v_2) = 0\}$
2. For every edge $\overline{e} \in \overline{E}$ we create a (unique) literal $l_{\overline{e}} \in \mathfrak{L}$
   $\mathfrak{L} : \overline{E} \to \mathfrak{L}$ is (additionally) a function, mapping edges to literals: $\mathfrak{L}(\overline{e}) = l_{\overline{e}}$
3. For every vertex $v \in V$ we create a set of literals $\wp_v$ (a prime implicant) by:
   $\wp_v = \left\{l_{\overline{e}_i} \in \mathfrak{L} \,|\, \exists \overline{e}_i \in \overline{E} : (\mathfrak{L}(\overline{e}_i) = l_{\overline{e}_i} \wedge \exists v' \in V : \{v', v\} = \overline{e}_i)\right\}$
4. We define for every $\wp_v$ the requirements maximum as $P_{req}(\wp_v) = 0.1$ (or any other constant with a value $\epsilon < 1$).
5. The prime implicant to be assessed is given by the complete set of literals: $\wp^? = \mathfrak{L}$

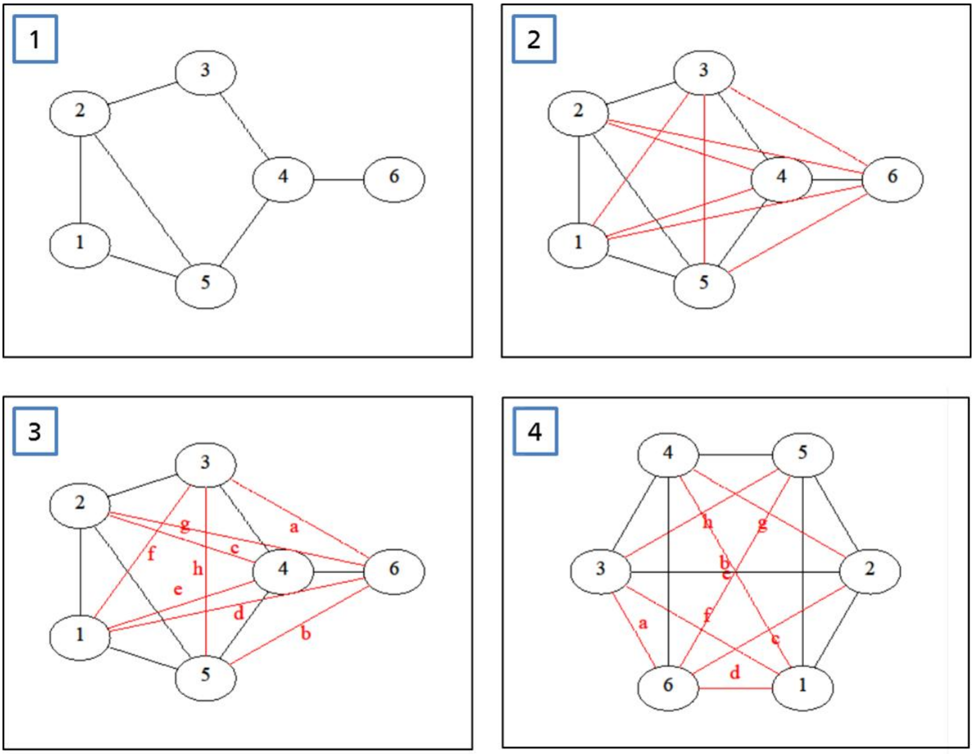


Figure 3.55: Example showing the clique problem reduction

We claim that the result of our knowledge extraction function 3.84 selects a set of disjoint prime implicants that imply the maximum clique in the above graph. Before proving our claim, we show a small example (cf. Figure 3.55).

The four sub-figures in Figure 3.55 show in (1) the original graph for which we want to find the maximum clique for. In (2), we extended in red the missing edges (cf. step 1. from above). In (3), we added (unique) literals to each edge (cf. step 2. above). The final set of literals $\mathfrak{L} = \{a, b, c, d, e, f, g, h\}$. In (4), we rearranged the graph to better identify the labels of the edges. The next step, 3., from the above results is a list of sets of literals and 4. assigns a safety requirement of $P_{req} = 0.1$ to it. Both are shown in Table 3.5.

The prime implicant to be analyzed is given by $\wp^? = \mathcal{L} = \{a, b, c, d, e, f, g, h\}$. In Table 3.6, we listed the set of all disjoint sets and calculated the respective requirements.

The minimum in Table 3.6 is given in the last row (with ID=14). The sets of literals belong to the graph vertices 1, 2, and 5. This is also the maximum clique set.

v	$\wp_v$	$P_{req}(\wp_v)$
1	$\{d, e, f\}$	0.1 $(= \epsilon)$
2	$\{c, g\}$	0.1 $(= \epsilon)$
3	$\{a, f, h\}$	0.1 $(= \epsilon)$
4	$\{e, g\}$	0.1 $(= \epsilon)$
5	$\{b, h\}$	0.1 $(= \epsilon)$
6	$\{a, b, c, d\}$	0.1 $(= \epsilon)$

Table 3.5: Clique problem translation steps 3.+4.

ID	disjoint set of sets of literals	$P_{req}(\wp^?)$
1	$\{d, e, f\}$	0.1 $(= \epsilon)$
2	$\{c, g\}$	0.1 $(= \epsilon)$
3	$\{a, f, h\}$	0.1 $(= \epsilon)$
4	$\{e, g\}$	0.1 $(= \epsilon)$
5	$\{b, h\}$	0.1 $(= \epsilon)$
6	$\{a, b, c, d\}$	0.1 $(= \epsilon)$
7	$\{d, e, f\}, \{c, g\}$	0.01 $(= \epsilon^2)$
8	$\{d, e, f\}, \{b, h\}$	0.01 $(= \epsilon^2)$
9	$\{c, g\}, \{b, h\}$	0.01 $(= \epsilon^2)$
10	$\{c, g\}, \{a, f, h\}$	0.01 $(= \epsilon^2)$
11	$\{a, f, h\}, \{e, g\}$	0.01 $(= \epsilon^2)$
12	$\{e, g\}, \{b, h\}$	0.01 $(= \epsilon^2)$
13	$\{e, g\}, \{a, b, c, d\}$	0.01 $(= \epsilon^2)$
14	$\{d, e, f\}, \{c, g\}, \{b, h\}$	0.001 $(= \epsilon^3)$

Table 3.6: Determination of minimal requirement based on disjoint sets of literals

Proof:

1. We prove that each disjoint set of prime implicants induces a clique in its respective set of vertices:
   We assume that there is a disjoint set of prime implicants P that does not induce a clique. The respective vertices are $V_P \subseteq V$. Without lack of generality, let$v_1, v_2 \in V_P$ be the vertices without an edge in the original graph, and $\wp_1, \wp_2 \in P$ the respective prime implicants. By construction, we include the missing edge and derive a unique literal l for it. We further know that this literal must be in $\wp_1$ as well as in $\wp_2$. This, however, is a contradiction to our assumption that the set of prime implicants P is disjoint.
   □

2. We prove that the minimal requirement results in the maximum clique:
   Each set of literals in the disjoint set of prime implicants has the same

requirement (0.1). We can in this case define that the actual requirement of a set of disjoint prime implicants $P_{\wp}$ is $P_{act}(P_{\wp}) = 0.1^{|P_{\wp}|}$. Due to the fact that each set of literals "belongs" to one vertex in the graph G, it is obvious that the induced clique has the order $|P_{\wp}|$. Using this observation, the rest of the proof is trivial. □

■

This tells us that our optimization problem is at least *NP-equivalent*. Due to good tool support and efficient heuristics for optimizations, we decided to use linear programming to implement our knowledge extraction function. A linear program is a problem that can be expressed in the following canonical form:

$$\begin{aligned} \text{maximize } & c^T x \\ \text{subject to } & Ax \leq b \\ \text{and } & x \geq 0 \end{aligned}$$

A binary integer program (BIP) is a program in which the variables x can only hold 0 or 1. This problem is classified as NP-hard.

We can translate our problem into an BIP. The construction of matrix A is as follows:
Each column can be thought of as being one prime implicant we already assessed and each row describes a literal in the prime implicant we are looking for. The concrete entry in the table (the coefficient) is 0 or 1, dependent on whether the literal is contained in the prime implicant or not. The vector b has only 0 and 1 entries as well. It "selects" the relevant literals, i.e., if one literal is contained in the prime implicant we are looking for, b has as the respective entry: "1" and "0" otherwise.

$$b(\wp, l) = \chi_{lit}(\wp, l) := \begin{cases} 1 & \text{if } l \in \wp \\ 0 & \text{else} \end{cases} \tag{3.88}$$

For the optimization function $\max\{c^T x\}$ we reformulate our knowledge extraction minimization function:

$$\min\left\{\prod_{\wp\in\mathcal{P}^?} P_{req}(\wp)\right\} \Leftrightarrow \tag{3.89}$$

$$\min\left\{\log\left\{\prod_{\wp\in\mathcal{P}^?} P_{req}(\wp)\right\}\right\} \Leftrightarrow \tag{3.90}$$

$$\min\left\{\sum_{\wp\in\mathcal{P}^?} \log\left\{P_{req}(\wp)\right\}\right\} \Leftrightarrow \tag{3.91}$$

$$\max\left\{\sum_{\wp\in\mathcal{P}^?} -\log\left\{P_{req}(\wp)\right\}\right\} \tag{3.92}$$

Thus, if we assume (w.l.o.g) that $\mathcal{P}^{\surd}$ has a numbering such as: $\mathcal{P}^{\surd} = \{\wp_1, \wp_2, \ldots, \wp_n\}$, then:

$$\vec{c}_{ARID} = \begin{pmatrix} -\log\left\{P_{req}(\wp_1)\right\} \\ -\log\left\{P_{req}(\wp_2)\right\} \\ \vdots \\ -\log\left\{P_{req}(\wp_n)\right\} \end{pmatrix} \qquad \vec{x}_{ARID} = \begin{pmatrix} x_{\wp_1} \\ x_{\wp_2} \\ \vdots \\ x_{\wp_n} \end{pmatrix} \tag{3.93}$$

If we assume the literals $l_1, l_2, \ldots, l_k$ and we are investigating for prime implicant $\wp$, we get:

$$\vec{b}_{ARID} = \begin{pmatrix} \chi_{lit}(\wp, l_1) \\ \chi_{lit}(\wp, l_2) \\ \vdots \\ \chi_{lit}(\wp, l_k) \end{pmatrix} \tag{3.94}$$

$$A_{ARID} = \begin{array}{c} \\ l_1 \\ l_2 \\ \vdots \\ l_k \end{array} \begin{array}{c} \begin{array}{cccc} \wp_1 & \wp_2 & \cdots & \wp_n \end{array} \\ \begin{pmatrix} \chi_{lit}(\wp_1, l_1) & \chi_{lit}(\wp_2, l_1) & \ldots & \chi_{lit}(\wp_n, l_1) \\ \chi_{lit}(\wp_1, l_2) & \chi_{lit}(\wp_2, l_2) & \ldots & \chi_{lit}(\wp_n, l_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{lit}(\wp_1, l_k) & \chi_{lit}(\wp_2, l_k) & \ldots & \chi_{lit}(\wp_n, l_k) \end{pmatrix} \end{array} \tag{3.95}$$

Thus the ARID prime implicant BIP problem is given by:

$$\max\left\{\vec{c}_{ARID}{}^{T}\,\vec{x}_{ARID}\right\} \text{ with } A_{ARID}\vec{x}_{ARID} \leq \vec{b}_{ARID} \tag{3.96}$$

As described above, the vector $\vec{b}_{ARID}$ can contain entries with 0. However, this means that the literal cannot be contained in the solution vector. Thus, we can reduce the problem by removing all prime implicants (columns) from A_{ARID} for which the literal (the row) should not be contained (as indicated by $\vec{b}_{ARID}$). Removing a column does however, require to removing the variable from the solution vector as well. In the following concrete example, we do not construct the complete BIP but only the reduced version, as discussed above.

Example

Figure 3.56 shows an abstract system consisting of four functions (f1, f2, f3, f4) displayed as black rectangles. The detailed information displays Fault Trees for four service failures (sf1, sf2, sf3, sf4) and the respective failure logic. The basic event set B is given as $B = \{a, b, c, d, e, f\}$.

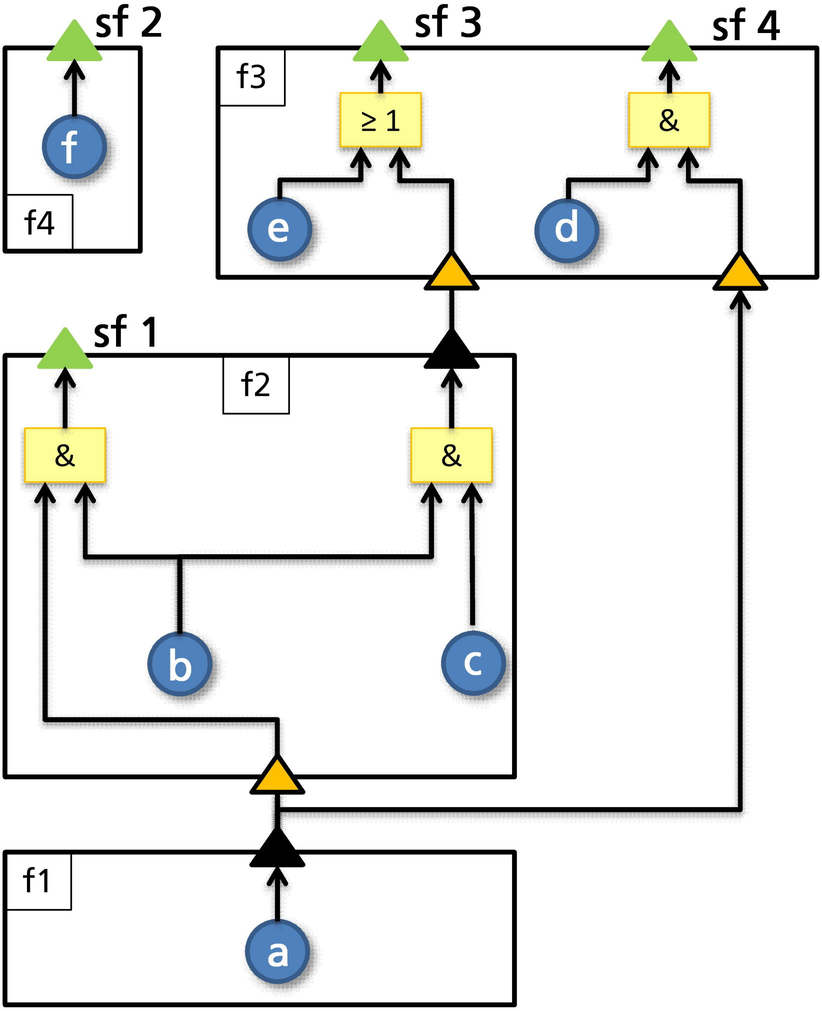


Figure 3.56: Abstract example to show the knowledge extraction idea of ARID

We assume that the hazards h with single service failures have already been assessed. The results of these assessments are shown in Table 3.7. The first column indicates the assessed service failure, the second column contains the result as probability, the third column shows the failure logic, and the fourth column represents the Boolean logic formula.

Table 3.7: Result of the four single service failure hazard and risk assessments.

$h =$	$P_{req}(\{h\}) =$	$f_{FT}(h) =$	$\mathcal{P}(h) =$
$\{sf1\}$	0.76%	$a \wedge b$	$\{\{a, b\}\}$
$\{sf2\}$	0.076%	f	$\{\{f\}\}$
$\{sf3\}$	7.6%	$b \wedge c \vee e$	$\{\{a\}, \{b, c\}\}$
$\{sf4\}$	53%	$a \wedge d$	$\{\{a, d\}\}$

Thus, the internal knowledge we can derive from these assessments is given by:

Table 3.8: Internal knowledge of prime implicant probability requirements

$\wp$	$P_{req}(\wp)$
$\{b, c\}$	7.6%
$\{e\}$	7.6%
$\{a, b\}$	0.76%
$\{a, d\}$	53%
$\{f\}$	0.076%

Now we want to know whether we need to assess the hazard $h_{\{3,4\}} = \{sf3, sf4\}$ consisting of the two service failures $sf3$ and $sf4$. We therefore need to create the combined Boolean logic formula:

$$sf3 \wedge sf4 = (b \wedge c \vee e) \wedge (a \wedge d) = \overbrace{(a \wedge d \wedge e)}^{\wp_1^? = \{a,d,e\}} \vee \overbrace{(a \wedge b \wedge c \wedge d)}^{\wp_2^? = \{a,b,c,d\}}$$

Both prime implicants $\wp_1^?, \wp_2^?$ are unknown. We use our binary logic programming approach to determine the currently assurable probability.
The BIP for $\wp_1^? = \{a, d, e\}$:

$$A = \begin{array}{c} \\ a \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccc} \{e\} & \{a,b\} & \{a,d\} \end{array} \\ \left(\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{array} \right) \end{array} \quad \vec{x} = \begin{pmatrix} x_e \\ x_{a,b} \\ x_{a,d} \end{pmatrix} \quad \vec{c} = \begin{pmatrix} 1.119 \\ 2.119 \\ 0.276 \end{pmatrix} \quad \vec{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

The solution of this BIP is:

$$\vec{x_{sol}} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \qquad P_{req}(\wp_1^?) = 0.05776\%$$

For this prime implicant we can assure a probability of 0.05776%!

The BIP for $\wp_2^? = \{a, b, c, d\}$:

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{ccc} \{b,c\} & \{a,b\} & \{a,d\} \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \qquad \vec{x} = \begin{pmatrix} x_{b,c} \\ x_{a,b} \\ x_{a,d} \end{pmatrix}$$

$$\vec{c} = \begin{pmatrix} 1.119 \\ 2.119 \\ 0.276 \end{pmatrix} \qquad \vec{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The solution of this BIP is:

$$\vec{x_{sol}} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad P_{req}(\wp_2^?) = 4.028\%$$

For this prime implicant we can only assure a minimal probability of 4.028%!

The final step to decide whether the hazard $h_{\{3,4\}}$ needs to be assessed is to calculate the maximum of the prime implicant's assurable probabilities. This is obviously 4.028%. We know that in the automotive domain, the strictest safety requirement has ASIL D, an associated failure rate of 10^{-8}, and a probability of 0.076%. Thus, we cannot exclude this hazard from the assessment.

Decision Optimization

In the example above, we discussed a hazard consisting of the two service failures $sf3$ and $sf4$. While the process via a binary linear program is always possible, its complexity is exponential. In some special cases we can reduce the decision complexity and derive the result much easier. The improvement possibility lies in the dependence of hazards (or service failures). Due to the fact that we represent all fault trees as disjunctions of conjunctions of literals, we can define three internal dependency cases.

1. totally independent
2. sharing a prime implicant
3. sharing a subset of a prime implicant (including sharing only one basic event)

These three cases are schematically shown in Figure 3.57. Each dependency case has some inherent knowledge we can use to optimize knowledge extraction. The "summary" of the knowledge is shown as a formula at the top of the figure. In the following, we will present the derivation of this for each case in the following.

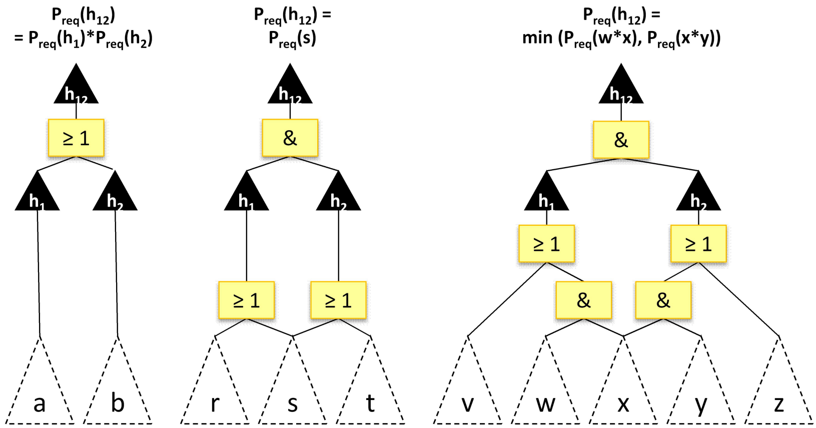


Figure 3.57: Figure showing the three possibilities for internal dependencies

The leftmost case shows the independence case. Let us come back to our example (cf. Figure 3.56). If we want know whether we need to analyze the hazard $h_{\{3,4\}} = \{sf2, sf3\}$, we see that the two service failures do not share any basic event; thus, they are internally independent of each other. The prime implicants to be investigated are:

$$sf2 \wedge sf3 = \overbrace{(b \wedge c \wedge f)}^{\wp_1^? = \{b,c,f\}} \vee \overbrace{(e \wedge f)}^{\wp_2^? = \{e,f\}}$$

The BIP matrix A_{ARID} for the prime implicant $\wp_1^?$ looks like this:

$$A = \begin{array}{c c} & \begin{array}{cc} \{b,c\} & \{f\} \end{array} \\ \begin{array}{c} b \\ c \\ f \end{array} & \left(\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{array} \right) \end{array}$$

The obvious solution is given by "selecting" both prime implicants $\{b, c\}, \{f\}$. Generalizing the independence case yields the following:
We assume two (internally independent) service failures sf_x, sf_y. Both have a corresponding set of prime implicants in the set of assessed prime implicants $\mathcal{P}^{\surd}$. Due to the internal independence, there is no literal l_i that occurs in a prime implicant from sf_x as well as one of sf_y. We can therefore separate both sets $\mathcal{P}^{\surd}$ and $\mathfrak{L}$ into disjoint sets of prime implicants and literals

each belonging to either sf_x or sf_y. Without lack of generality, we assume that the first subset $\mathcal{P}_x^{\surd} = \{\wp_1, \ldots, \wp_k\}$ and $\mathfrak{L}_x = \{l_1, \ldots, l_m\}$ belongs to sf_x and the second subset $\mathcal{P}_y^{\surd} = \{\wp_{k+1}, \ldots, \wp_n\}$ and $\mathfrak{L}_y = \{l_{m+1}, \ldots, l_o\}$ belongs to sf_y. The matrix $A_{_{ARID}}$ always looks like this:

$$A_{_{ARID}} = \begin{array}{c|c|c} & \wp_1 \cdots \wp_k & \wp_{k+1} \cdots \wp_n \\ \hline l_{m+1} \ldots l_o & 0 & A_{_{ARID}}(sf_y) \\ \hline l_1 \ldots l_m & A_{_{ARID}}(sf_x) & 0 \end{array}$$

Thus, we have two independent BIPs calculating the actual requirement of sf_x and sf_y. This we know already because we already assessed the risk of both single service failures. The result is the multiplication of both top-level safety requirements: $P_{req}(f_{FT}(sf_x)) * P_{req}(f_{FT}(sf_y))$.
$\Rightarrow$ We can omit the construction of a BIP if the service failures are internally independent. However, we get the same result by simply multiplying the top-level requirements.

The middle case of dependency shown in Figure 3.57 is given by the fact that the two hazards share a complete prime implicant $\wp_s$. Due to the fact that

$$(\wp_s \Rightarrow f_{FT}(h_1)) \wedge (\wp_s \Rightarrow f_{FT}(h_2)) \leftrightarrow \wp_s \Rightarrow (f_{FT}(h_1) \wedge f_{FT}(h_2)) \quad (3.97)$$

$\wp_s$ is also a prime implicant of the conjunction of the two hazards (or the "new" hazard h_{12}). All other prime implicants are mutually independent (of h_1 and h_2) and can therefore be treated like the first case: Determine the actual requirement by multiplication. Due to the fact that the final decision about the assurable actual requirement of h_{12} is determined by determining the maximum over all prime implicant requirements, we can state that the maximum requirement is given by the shared prime implicants requirement. This is easy to understand and we therefore omit proving this. We rather want to point out that for deciding whether two functions share (only) a prime implicant, we can pair-wise compare the prime implicants yielding a complexity of $\mathcal{O}(n^2)$.

The last case (the rightmost case in Figure 3.57) is given by sharing only a subset of a prime implicant. This means that one prime implicant of h_{12} in the figure would be $\wp_{wxy} = w \wedge x \wedge y$. For both single hazards, however, this is not a prime implicant (but only an implicant). If we analyze (via the BIP) this prime implicant $\wp_{wxy}$, two solutions for the implied constraint system of the BIP ($A * x \leq b$) are obviously the two prime implicants $\wp_{wx} = w \wedge x$ and $\wp_{xy} = x \wedge y$. However, this information is not very valuable because this basically yields that the actual requirement of the conjunction is the minimum of the actual requirement of the single hazards.

As stated above, all three cases are a kind of heuristics as to whether it makes sense to construct and evaluate BIPs. The decision is given by evaluating whether the actual requirement is lower than the minimally achievable requirement boundary (as discussed above). The least restrictive case is obviously the last case. From this case we can derive that we do not need to assess combinations of hazards if one hazard already reaches the boundary. More generally, all hazards that contain a partial set of service failures that have reached the requirements boundary do not need to be analyzed anymore!

Up to now we dealt with optimizing the efficiency and preciseness of the prediction of the already assurable requirements boundary of a hazard. This was based solely on internal information, such as information on error propagation and prime implicant requirements. We can use external knowledge as well. We cannot use external knowledge to better predict the assessment result of a hazard, but we can try to lower our "expectations". By expectations we mean that we have assumed up to now that each assessment can result in the (maximal) domain requirement (or the respective minimal probability). If we could reduce our expectations regarding possible risk assessment results, we could sort out hazard analyses even more effectively!

The starting point for this are the operational situations. If we consider a set of service failures, the service failures must be compatible in the sense that they must be critical in a common situation. If one service failure is, for example, only critical if the vehicle is braking and another one is only critical if the vehicle is accelerating, there is no critical common situation, and the hazard does not need to be assessed.

OASIS and HEAT dealt with the formalization and exposure assessment support of operational situations. In the following, we will recap the part of the respective discussions that are relevant for ARID.

Relevant "Operational Situation" part

The necessary part of (GOBI and) OASIS/HEAT for ARID is the formal representation of situations, their influence factors and influence factor dependencies (as domain knowledge). The formal part we need in ARID are the set of influence factors, *IF*, the set of situations *SIT*, and their relations "*imply*, *influence*, and *exclude*" (cf. 3.3.3):

$$SIT = P(IF) \text{ or } sit \subseteq IF \text{ with } sit \in SIT \tag{3.98}$$

With respect to the introduced relations capturing domain knowledge, of special interest for ARID is the information whether situations are mutually exclusive or not (please recall the "*exclude*"-relation). If we query the concrete model for all influence factors that are mutually exclusive, we get a list of pairs of influence factors. Let us assume that we have an "incompatibility" relation that extracts this information:

$$R_{\text{if-incomp}} \subseteq IF \times IF \text{ with} \tag{3.99}$$

$$(if_a, if_b) \in R_{\text{if-incomp}} \Leftrightarrow if_a \textit{ excludes } if_b \tag{3.100}$$

This means that a tuple (if_a, if_b) denotes that the influence factor if_a cannot occur together with if_b in one situation. An example would be (*highway, country road*) - one can obviously only drive either on a highway or on a country road. With this relation we can define the (in-)compatibility of situations, by introducing the decision function $\chi_{\text{sit-comp}}$ as follows:

$$\chi_{\text{sit-comp}} : SIT \times SIT \to \{0,1\} \text{ with} \tag{3.101}$$

$$\chi_{\text{sit-comp}}(sit_a, sit_b) := \begin{cases} 0 \text{ if } \exists\,(if_a, if_b) \in R_{\text{if-incomp}} \text{ with} \\ \qquad if_a \in sit_a \wedge if_b \in sit_b \\ 1 \text{ else} \end{cases} \tag{3.102}$$

This information can be used to "join" situations. For this we overload the usual mathematical operator $\cup$ and mark this special operator with a tilde: $\widetilde{\cup}$; hence we define:

$$sit_a \widetilde{\cup} sit_b = \begin{cases} \emptyset & \text{if } \chi_{\text{sit-incomp}}(sit_a, sit_b) = 1 \\ sit_a \cup sit_b & \text{else} \end{cases} \tag{3.103}$$

A situation "join" can be thought of as integrating two sets of influence factors into one situation.
Analogously, we use $\widetilde{\bigcup}$ to "join" sets of situations. We can think of this set-join as an element-wise situation join:

$$\widetilde{\bigcup}\{sit_1, sit_2, \ldots, sit_n\} = sit_1 \,\widetilde{\cup}\, sit_2 \,\widetilde{\cup}\, \ldots \,\widetilde{\cup}\, sit_n \tag{3.104}$$

This can now be used for deciding whether a set of service failures can share a "common" situation. We assume in ARID that the single service failure analyses have been done. Thus, we know for each service failure sf the set of relevant situations. For ARID we define a "retrieval" function:

$$f_{\mathsf{relSit}_{SF}} : SF \to P(SIT) \text{ with} \tag{3.105}$$

$$f_{\mathsf{relSit}_{SF}}(sf) \;= \{sit \in SIT \,|sit \text{ is relevant for } sf\} \tag{3.106}$$

If we transition from one service failure to hazards in general, i.e., to sets of service failures, we can define a similar function working with hazards instead of single service failures (leveraging our above-introduced situation "join"):

$$f_{\mathsf{relSit}_H} : H \rightarrow P(SIT) \text{ with} \tag{3.107}$$

$$\begin{aligned} f_{\mathsf{relSit}_H}(h) = \{ & sit_1 \mathbin{\widetilde{\cup}} \ldots \mathbin{\widetilde{\cup}} sit_n \mid \\ & \{sit_1 \in SIT_1 \wedge \ldots \wedge sit_n \in SIT_n\} \wedge \\ & \quad \forall i, j \in [1..n] : i \neq j \Rightarrow \chi_{\text{sit-comp}}\left(sit_i, sit_j\right) = 1 \ \} \end{aligned} \tag{3.108}$$

with

$$\begin{aligned} h &= \{sf_1, \ldots, sf_n\} \text{ and} \\ SIT_1 &= f_{\mathsf{relSit}_{SF}}(sf_1) \\ &\vdots \\ SIT_n &= f_{\mathsf{relSit}_{SF}}(sf_n) \end{aligned}$$

This definition gives us for each hazard $h \in H$ the set of relevant situations. This set is empty if the service failures are not dangerous at a common time.

In the section HEAT 3.4, we presented a CBR approach for supporting the safety engineer in the task of assessing new situations. We used a partial order to define consistent assessments of situations. We reuse this in ARID as well:
The above -etermined set of situations can be pre-processed by HEAT to determine the maximal exposure occurrence. Let us assume that the maximal exposure of the set of joined situations is "E2". If we look up all the possible ASIL values we could determine, we will recognize that with "E2" we can only reach "ASIL B". This does, however, reduce the expectable assessment result and thereby the boundary beyond which we do not need to assess the hazard anymore. Thus, by leveraging OASIS together with HEAT, we can increase the effectiveness of reasoning about unnecessary hazard assessments.

Relevant "Controllability" part

As discussed as part of the Human FVM section in GOBI 3.2.2, a controllability argument usually contains information about other services that are used in the controllability action. An example is that a service failure leading to a hazard which affects lateral acceleration might be controllable by applying an inverse momentum via the steering wheel. Of course, this is possible if and only if the steering capability is still available. If we have a common cause and unintentional lateral acceleration fails together with the steering capability, we cannot argue with this kind of controllability.
A controllability analysis therefore results in a controllability argument $ca \in CtrlArg$. We can define that a controllability analysis is a function f_{Ctrl} with a hazard and an operational situation as input and the controllability

argument *ca* as output:

$$
\begin{aligned}
f_{\mathsf{ctrl}} : H \times SIT \rightarrow CtrlArg \text{ with} && (3.109)\\
f_{\mathsf{ctrl}}\left(h, sit\right) \qquad = \left\{ca \in CtrlArg \left| h \in H \wedge sit \in SIT\right.\right\} && (3.110)
\end{aligned}
$$

For ARID, the dependency of services is particularly relevant, as described above. Thus, we are interested in the required service. For this we define another function f_{ctrlFcts} and concatenate this function with our function f_{ctrl}:

$$
\begin{aligned}
f_{\mathsf{ctrlFcts}} : CtrlArg \qquad \rightarrow P\left(S\right) && (3.111)\\
f_{\mathsf{fctReliance}} : H \times SIT \rightarrow P\left(S\right) \text{ with} && (3.112)\\
f_{\mathsf{fctReliance}}\left(h, sit\right) \qquad := f_{\mathsf{ctrlFcts}}\left(f_{\mathsf{ctrl}}\left(h, sit\right)\right) && (3.113)
\end{aligned}
$$

3.5.3 ARID Analysis Process

In the previous sections of ARID, we discussed the basic modeling assumptions and presented the basic idea of ARID (cf. 3.5.1). In section 3.5.2, we discussed the possibilities of predicting the result of a risk assessment in order to automatically reduce the number of hazards to be assesses.

In this section, we put the basics and the ARID prediction concepts together and propose a process that efficiently handles the (potentially) exponential number of hazards to be assessed.

In order to understand our process proposal, we remind that the reasons for considering hazards consisting of multiple service failures in the HRA were:

1. Due to the ever increasing integration and interaction of functions, we cannot assume that functions will fail independently.
2. Due to the reduction of ASILs by controllability arguments that rely on other functions, these functions are dependent and one should investigate the case in which the function and the controllability-relevant function fail at the same time.

While the case of assessing totally independent functions (theoretically) exists as well, it implies a multiple point failure, which many industry standards assume as safe. To be complete with respect to the $2^{|sf|}$ hazards, we include this case as well, with an explicit comment on its practical relevance.

Accordingly, we structured the ARID analysis process into four steps:

1. Analyze single service failures
   This is the step that is performed "traditionally".
2. Analyze sets of internally dependent service failures
   This step extends the first step by considering multiple service failures that are internally dependent ($\approx$ common cause). This case is addressed in ISO 26262 as "multifunctional degradation". It is therefore even implied by an international standard.
3. Analyze sets of externally dependent service failures
   Besides internal dependencies, we identified external dependencies via controllability arguments. We therefore cannot assume that a common failure of a service and its "controllability" service is simply the conjunction of the single (internally independent) hazards. Due to the reduction of the ASIL via decreased C-values ($=$ controllability arguments), we might end up in a higher ASIL!
4. Analyze sets of independent service failures
   Even though the services are totally independent, one must ensure their independence. This level is therefore important for development because it could potentially lead to higher requirements for the separation of the (independent) functions.

In the following process, we will use the following assumptions and conventions:
(1) All hazards that have been analyzed already are in the set $H_{analyzed}$. For each hazard $h \in H_{analyzed}$ we assume that:

- the function f_{relSit_H} delivers the relevant situations for the hazard
- the function $f_{\mathsf{fctReliance}}$ delivers the service dependency used in the controllability argument.

(2) We know for each analyzed hazard (in $H_{analyzed}$), the safety requirement in terms of minimal probability $\mathcal{P}_{req}$ and a minimal assessment boundary. (Please note that the minimal boundary given by the domain is the minimum minimal assessment boundary.) As described in 3.5.2, we can increase (relax) this boundary using the exposure value of previously analyzed hazards.
We can derive the minimal assessment boundary for an assessed hazard h by:

$$f_{\mathsf{max\ E}}(h) = \max\{exposure(sit) \mid sit \in f_{\mathsf{relSit}_H}(h)\} \tag{3.114}$$

$$f_{\mathcal{P}_{min}}(h) = \begin{cases} 0,076\% & \text{if } f_{\mathsf{max\ E}}(h) = E4\ (\lambda = 10^{-8}, a = 8,7) \\ 0,76\% & \text{if } f_{\mathsf{max\ E}}(h) = E3\ (\lambda = 10^{-7}, a = 8,7) \\ 7,3\% & \text{if } f_{\mathsf{max\ E}}(h) = E2\ (\lambda = 10^{-6}, a = 8,7) \\ 53\% & \text{if } f_{\mathsf{max\ E}}(h) = E1\ (\lambda = 10^{-5}, a = 8,7) \end{cases} \tag{3.115}$$

(3) If one hazard exceeds its assessment boundary, we put all hazards that are super-sets of the analyzed hazard in the set of analyzed hazards. Please note that this might lead to an exponential *reduction* of hazards to be analyzed.

(4) Let $\nu : 2^{BE} \to 2^{|BE|}$ be a valuation function that assigns to each variable in BE a binary value $\in \{0,\ 1\}$. Using this valuation of Boolean variables, we can bound variables in Boolean logic formulas, such as our fault tree formula: $f_{FT}(sf)|_{\nu}$.

Analyze Single Service Failures

In this step, we assume that the traditional HRA is performed. We have linear ($\mathcal{O}(n)$) complexity in this step.

After this step, $H_{analyzed}$ consists of all hazards h with $|h| = 1$.

Analyze Sets of Internally Dependent Service Failures

This step deals with the analysis of internally dependent failures. Internally dependent means that the dependent service failures share at least one common basic event (assuming a fault tree representation of the error propagation). The challenge of a possibly exponential number of basic events is given in this step. Depending on the error propagation, we can assume that in a worst-case scenario, every service failure is dependent on every other

service failure.
We therefore separate this case by classifying basic events into single point faults and multiple point faults (leading to a multiple service failure). The rationale for this separation has two aspects. The first aspect is practical relevance. It is always important to take care of single point faults, while multiple point faults are sometimes omitted. The second, more scientific reason is that we know that all single fault cases already have a certain requirement assigned (from the first process step), while we cannot argue this for multiple point cases. The difference therefore lies in the decision process whether a further assessment is needed or not.

The single point fault case (the first case) is the case where one basic event leads to the triggering of multiple service failures. We therefore need a common cause analysis based on basic events. We term the resulting set the *maximal influence set* (mi). It is an influence set because the basic event (by definition a single point fault) triggers the top events in that set. We additionally qualified it to be maximal to ensure that ALL top events that are triggered are in the set!
More formally we can define:

$$mi(be) = \big\{sf \in SF \mid f_{FT}(sf)|_{\nu(\{be\})} = 1 \text{ with } \nu(\{be\}) = \{1\}\big\} \tag{3.116}$$

Please note that we denote with $f_{FT}(sf)|_{be=1}$ the fault tree belonging to the service failure sf in which the basic event be is bound to have the value "true $(= 1)$".
The final decision $f_{FT}(sf)|_{be=1} = 1$ can be efficiently decided using the internal Boolean logic representation of the service failure as a binary decision diagram (BDD): The BDD should collapse (with $|_{be=1}$) to the identity node $(= 1)$.

Once we know the set of maximal influence sets for each basic event, we apply our heuristic regarding whether further assessment is needed or not. If the result is that we need an assessment, the assessment is performed.

The second part of this internal dependency analysis addresses the situation in which a basic event can only trigger multiple top events if other basic events are present as well and no subset of them is able to trigger the same set of top events (= multiple point faults). Please note that this is the right case shown in Figure 3.57.

The difference in this case is that the basic event by itself does not trigger a service failure, but is one contributing factor. We can formalize this by:

$$f_{con} : BE \to 2^{SF} \text{ with} \tag{3.117}$$

$$f_{con}(be) = \{\ sf \in SF \mid \exists \text{ path from } be \text{ to } sf\ \} \tag{3.118}$$

$$= \{\ sf \in SF \mid \exists \nu,\ \exists \beta \subseteq BE \setminus \{be\} : |\beta| > 0 \land be \Rightarrow f_{FT}(sf)|_{\nu(\beta)}\ \} \tag{3.119}$$

Please note that the expression $|\beta| > 0$ removes all cases in which one basic event triggers a service failure, as in equation 3.116. (The case $|\beta| = 0$ would imply that the basic event triggers the service failure without any other variable restriction.)

Using the function f_{con} we can assign to each basic event a contribution importance. The value of the importance is given by the number of service failures to which the basic event contributes:

$$\textit{importance}(be) = |f_{con}(be)| \tag{3.120}$$

The higher the value, the more service failures are influenced and therefore the more important the basic event for system safety. We therefore use importance to steer the analysis process by "highest importance first".

The subsequent analysis step is performed for each basic event, starting with the basic event with the highest importance. The basic event be gives us a set of relevant service failures by $SF^{be} = f_{con}(be)$. We remove from this set all service failures that exceed the maximum requirements boundary (in the ARID analysis step 1).
Using this set, we successively create all sub-sets by their order. We start with the sub-sets that contain two service failures, continue with all supersets containing three service failures, and so on.
Thus, let $SF_n^{be} \subseteq SF^{be}$ with $\left|SF_n^{be}\right| = n$ denote the set of sets of service failures containing n service failures. We therefore start with SF_2^{be}. For each of these double service failures, we decide whether an assessment is needed. If an assessment is necessary, we do the assessment and store the result.
We continue with SF_3^{be} in the same way and move on to the next higher order SF_4^{be}. We continue the process until we reach the set containing all sets $SF_{|SF^{be}|}^{be}$.

This process step does, of course, have the potential of being exponential. From a safety engineering point of view, we point out that this complexity is self-made (by design): It is the system's complexity. We claim that if the design is highly coupled, the safety analysis is complex; if it is loosely coupled, the process terminates fast (because $\left|SF^{be}\right|$ is small). Thus, the complexity of this step simply reflects the system's complexity and is not a lack of analysis process capability!
A similar comment holds for the mathematical complexity w.r.t. time and space requirements for the computer on which the algorithm is running. The author argues that if it is not even possible to represent and work with the implied problem size in a current computer, we doubt that the system's complexity and thereby the system's safety is under control!

Analyze Sets of Externally Dependent Service Failures

In this step, we deal with externally dependent service failures. Please recall that external dependency is given by the controllability argument. If one service fails and the operator is able to avoid or mitigate harm by performing a counteraction, the service the operator relies on becomes a prerequisite for the safe functioning of the service under analysis (cf. 3.5.2).

We introduced in 3.5.2 the functional controllability reliance function $f_{fctReliance}$ (cf. 3.113), which delivers for each analyzed hazard the set of services on which the risk assessment is relying.

At this stage of the ARID process, all hazards consisting of either a single service failure or internally dependent service failures have been analyzed. Externally dependent service failures have not been analyzed yet as such, but some of them might already have been analyzed as part of the internal dependency analysis step. We therefore might have analyzed externally dependent service failures already! The remaining set of externally dependent service failures is the set of internally independent, but externally dependent service failures:

Let H be the set of all possible service failures. The relevant subset includes all hazards h for which it holds that:

- $h \notin H_{analyzed}$
  $\hookrightarrow$ The hazard has not been analyzed before.
- $\nexists h' \subseteq h \; : \; h' \in H_{analyzed} \wedge \mathcal{P}_{req}(h') \leq f_{\mathcal{P}_{min}}(h)$
  $\hookrightarrow$ The hazard is not a super-set hazard of a hazard h' that has already reached the minimal requirements boundary of the current hazard.
- $\nexists h' \subseteq h \; : \; f_{\mathsf{max\ E}}(h') > f_{\mathcal{P}_{max}}(h')$
  $\hookrightarrow$ There is no subset of service failures that restrict the exposure in such a way that the remaining assessment boundary is reached.
- $\exists h' \subseteq h \,|\, [\; \exists sit \in f_{\mathsf{relSit}_H}(h) \,|\, |f_{\mathsf{fctReliance}}(h, sit)| > 0 \;]$
  $\hookrightarrow$ The hazard is externally dependent.

Our proposed strategy is again to analyze the hazards consisting of two service failures and then increase the cardinality of the hazards to be analyzed.

This sounds very similar to the internal dependency case. We want to point out that the difference is that internal dependency is statically given by the system design and the failure propagation. External dependency (via controllability arguments) emerges from the analysis process itself. Due to the fact that the controllability (for automotive systems) needs to be very simple, the argumentation of risk reduction via controllability in cases of multiple service failures is very rare.

Analyze Sets of Independent Service Failures

This step finalizes the ARID analysis and makes the hazard analysis complete. The result of this step can deliver safety requirements for the separation mechanism of services or functions. The corresponding term in automotive industry is "freedom from interference".

If H is again the set of all possible hazards and we performed the previous steps completely, we can state that $H_{Ind} = H \setminus H_{analyzed}$ is the set that contains all hazards that contain independent partitions of service failures. The advantage is that we can apply the heuristic of totally independent hazards (cf. Figure 3.57). We can therefore multiply the actual requirements of the independent subsets of hazards. If we assume (the worst case) that each independent hazard has only ASIL A (thus, the minimal contribution for reaching the domain minimum), we can derive that after combining three independent ASIL A hazards, we already have the probability requirement of ASIL D!

$$\begin{aligned}
\text{ASIL A} \quad & \Rightarrow \mathcal{P}(8, 7a) = 0.0073 && (3.121)\\
\text{ASIL D} \quad & \Rightarrow \mathcal{P}(8, 7a) = 0.000076 && (3.122)\\
\hookrightarrow \quad & (0.0073)^x = 0.000076 && (3.123)\\
\hookrightarrow \quad & x * \ln(0.0073) = \ln(0.000076) && (3.124)\\
\hookrightarrow \quad & x = \frac{\ln(0.000076)}{\ln(0.0073)} \approx 2.744 && (3.125)
\end{aligned}$$

Thus, we can limit ourselves to the subset of independent hazards that are triple independent! If we encounter a higher original ASIL level, we can stop the analysis even faster. Thus, we have as worst case a cubic effort, i.e., $\mathcal{O}(n^3)$ in this step.

With this observation we finish our discussion of the ARID analysis process.

3.5.4 ARID Analysis Example

The previous sections of ARID presented our modeling, the fundamental idea, and a concrete analysis process. In this section, we will present an example used as an evaluation of ARID. Additionally, we will point out our tooling support for ARID.

ARID heavily relies on models and computer support for storing and processing of and reasoning on existing knowledge. As an evaluation of ARID, we developed a prototype tool and applied our analysis process to an example system using that tool. In the following, we will start by giving an overview of our tool support. Afterwards we will introduce the example system and at the end we will provide concrete results and screenshots of our analysis process.

The tool environment

In ARID section 3.5.1 we stated that we use a port-based system design and Component Fault Trees as failure logic model as the starting point for ARID. We consequently used the C²FT tooling environment. The basis of this environment is a standard UML tool called MagicDraw. The C²FT extension is realized as a UML profile and a plugin-based fault tree calculation backend.

The first step was to extend the meta-model by an ARID profile to represent the elements Hazard, Hazard Assesssment, Service Failure, Service, and Cutset.

Figure 3.58 shows the extension:

- At the very top, the *Hazard Assessment* element is shown. Hazard Assessment has as attributes:
  - a reference to the system to be analyzed (*itsSystem*)
  - an attribute that manages the references to all hazards (*Hazards*)
  - a variable that contains the current minimal probability of the domain (*DomainRequirement*). Thus, the domain requirement can be selected for each hazard assessment.
  - an attribute that contains references to all services of the system (*Service*)
- On the left, the element *Hazard* is shown. Hazard has the following attributes:
  - for easier handling in the ARID process, we store the order of the hazard in *Order*.
  - the variable *SafetyRequirement* contains the assigned requirement. This variable is only set if the hazard has been assessed.
  - the variable *ActualMaxRequirement*, in contrast, contains the currently assurable requirement. This variable basically holds the "knowledge" derivable for the respective hazard.

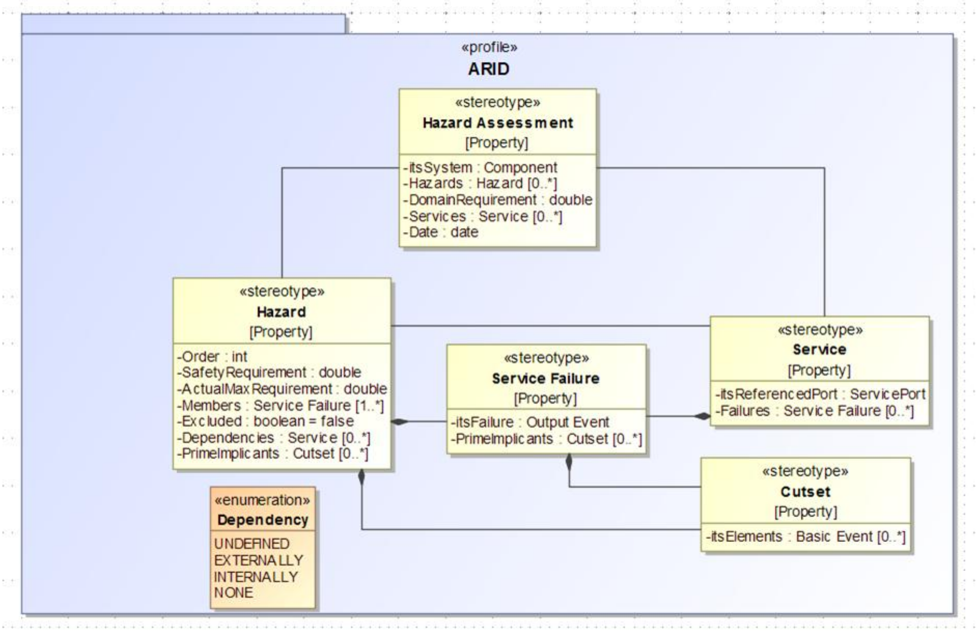

«profile»
ARID
«stereotype»
Hazard Assessment
[Property]
-itsSystem : Component
-Hazards : Hazard [0..*]
-DomainRequirement : double
-Services : Service [0..*]
-Date : date
«stereotype»
Hazard
[Property]
-Order : int
-SafetyRequirement : double
-ActualMaxRequirement : double
-Members : Service Failure [1..*]
-Excluded : boolean = false
-Dependencies : Service [0..*]
-PrimeImplicants : Cutset [0..*]
«stereotype»
Service Failure
[Property]
-itsFailure : Output Event
-PrimeImplicants : Cutset [0..*]
«stereotype»
Service
[Property]
-itsReferencedPort : ServicePort
-Failures : Service Failure [0..*]
«stereotype»
Cutset
[Property]
-itsElements : Basic Event [0..*]
«enumeration»
Dependency
UNDEFINED
EXTERNALLY
INTERNALLY
NONE

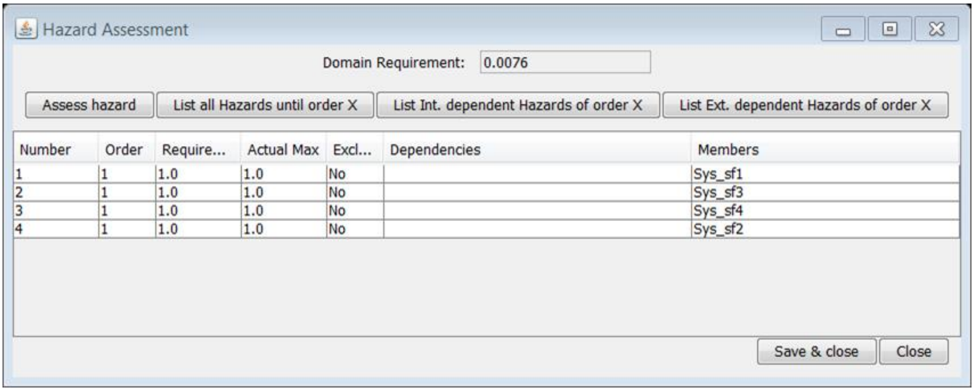
Figure 3.58: UML (MagicDraw) Profile for ARID

  - the *Members* of the Hazard are the service failures the hazard consists of.
  - the *Excluded* flag indicates whether a hazard has been excluded by the ARID decision algorithm.
  - the *Dependencies* indicate the external dependencies on other services.
  - the *PrimeImplicants* are obviously the prime implicants of the hazards.

- In the middle, the element *Service Failure* is shown. A service failure has the following attributes:
  - *itsFailure* is a reference to the respective top event in the Component Fault Tree
  - *PrimeImplicants* hold the prime implicants of the service failure (as implied by the Component Fault Tree).
- On the right, the element *Service* is shown with the following attributes:
  - *itsReferencedPort* is a reference to the port where the service is manifested.
  - one service might have multiple service failures; these are referenced/stored in the variable *Failures*.

- ▶ Finally, we modeled the prime implicants or, in this case, the cutsets explicitly with the element *Cutset*. A Cutset holds as attribute only a list of basic events *itsElements*.

This profile enables the model-based representation of the necessary elements as a MagicDraw model. The logic and the reasoning, however, is done in a Java-based ARID plugin. This plugin additionally provides an HRA assessment GUI, which enables easy assessment. This GUI is shown in Figure 3.59.


Hazard Assessment
Assess hazard
List all Hazards until order X
List Int. dependent Hazards of order X
List Ext. dependent Hazards of order X
Save & close
Close

Domain Requirement: 0.0076

Number	Order	Require...	Actual Max	Excl...	Dependencies	Members
1	1	1.0	1.0	No		Sys_sf1
2	1	1.0	1.0	No		Sys_sf3
3	1	1.0	1.0	No		Sys_sf4
4	1	1.0	1.0	No		Sys_sf2

Figure 3.59: ARID's main "control center": the assessment front-end

In order to check for correct knowledge handling, i.e., for requirements propagation and derivation of prime implicants, we additionally created a prime implicants graph view. This view shows the prime implicants of the system as a partial order graph. Figure 3.60 depicts this. The blue rectangles represent a prime implicant (cutset). The label shows the basic events belonging to the cutset in curly braces and behind that the current assessment knowledge the cutset holds. This knowledge is given by the current probability interval.

Since the readability of this graph is not optimal, and in order to "save" intermediate states of the graph, we created an export possibility to a GraphViz "dotty" graph (cf. "Dump to dot" button on the lower right of the Figure 3.60). Figure 3.61 shows the GraphViz-Dot version of the graph shown in Figure 3.60.

One implementation detail is that the ARID backend works as a free-running thread that monitors changes of the "knowledge base" and automatically (and asynchronously) optimizes the knowledge base in such a way that only relevant information is kept. Relevant information refers to prime implicants with a requirement that cannot be calculated/derived from other prime implicant knowledge. Figure 3.62 shows the status information of the opti-

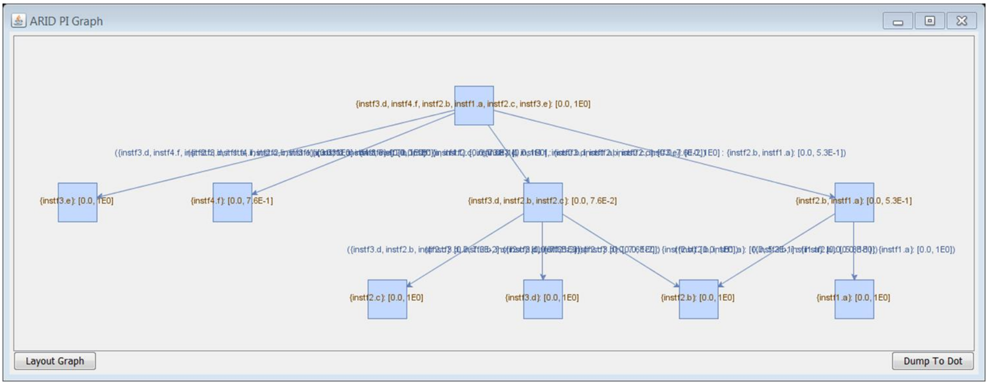

ARID PI Graph
{instf3.d, instf4.f, instf2.b, instf1.a, instf2.c, instf3.e}: [0.0, 1E0]
{instf3.e}: [0.0, 1E0]
{instf4.f}: [0.0, 7.6E-1]
{instf3.d, instf2.b, instf2.c}: [0.0, 7.6E-2]
{instf2.b, instf1.a}: [0.0, 5.3E-1]
{instf2.c}: [0.0, 1E0]
{instf3.d}: [0.0, 1E0]
{instf2.b}: [0.0, 1E0]
{instf1.a}: [0.0, 1E0]
Layout Graph
Dump To Dot

Figure 3.60: ARID's evaluation window showing the current "knowledge base" as prime implicant graph

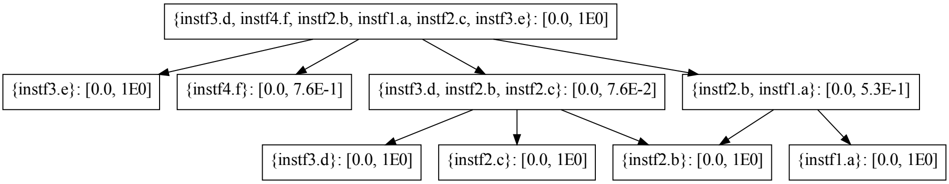

{instf3.d, instf4.f, instf2.b, instf1.a, instf2.c, instf3.e}: [0.0, 1E0]
{instf3.e}: [0.0, 1E0]
{instf4.f}: [0.0, 7.6E-1]
{instf3.d, instf2.b, instf2.c}: [0.0, 7.6E-2]
{instf2.b, instf1.a}: [0.0, 5.3E-1]
{instf3.d}: [0.0, 1E0]
{instf2.c}: [0.0, 1E0]
{instf2.b}: [0.0, 1E0]
{instf1.a}: [0.0, 1E0]

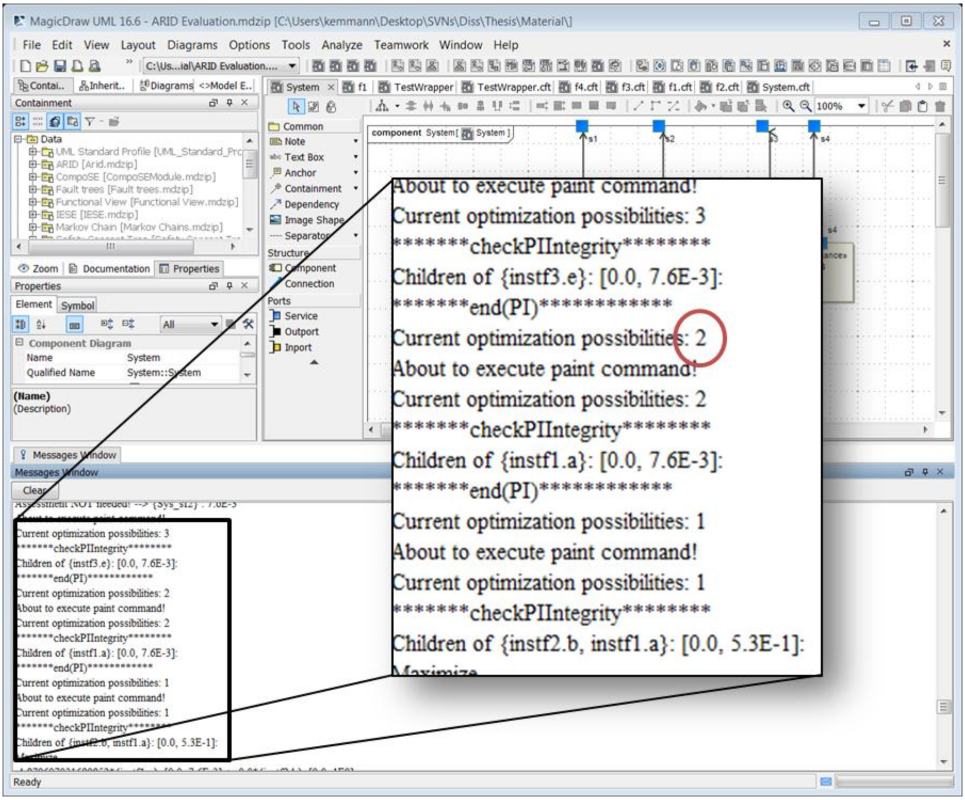
Figure 3.61: ARID's knowledge base exported to a GraphViz dot graph

mization thread posted to the message window of MagicDraw. The thread indicates via the message window how many optimizations are still possible and performs these optimizations as a "prime implicant integrity check" (cf. line "********checkPIIntegrity*********" in Figure 3.62).

We decided to integrate this decoupled optimization thread in order to avoid unnecessary complexity limitations because redundant or superfluent information is stored in the model. The design as an asynchronous thread was chosen because this is "only" an optimization thread. The assessment should not be stalled because internal model optimizations are running. This allows us to continue with the assessment even though optimizations are running in the background and even if not all optimizations have been executed yet.

Finally, the concrete assessment is performed/entered into an assessment dialog. A screenshot of this dialog is shown in Figure 3.63.

At the very top on the left, there is a text field for entering the hazard requirement, i.e., the quantitative assessment result. On the right next to this text field is a drop-down box for selecting the maximal situation the assessment has. Please recall that this was important for lowering of domain boundary based on the maximally expectable common situation! Finally, in



```
About to execute paint command!
Current optimization possibilities: 3
*******checkPIIntegrity********
Children of {instf3.e}: [0.0, 7.6E-3]:
*******end(PI)************
Current optimization possibilities: 2
About to execute paint command!
Current optimization possibilities: 2
*******checkPIIntegrity********
Children of {instf1.a}: [0.0, 7.6E-3]:
*******end(PI)************
Current optimization possibilities: 1
About to execute paint command!
Current optimization possibilities: 1
*******checkPIIntegrity********
Children of {instf2.b, instf1.a}: [0.0, 5.3E-1]:
Maximize
```

Figure 3.62: ARID's knowledge base optimization thread

the lower part, one can select the services on which this assessment relies. This is the ARID-relevant fraction of the controllability argument!

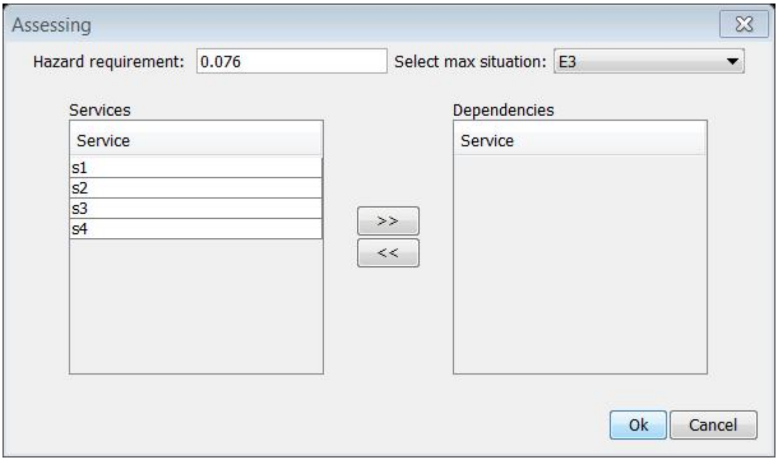
The Example System

The example system we used for evaluation was already introduced in section "3.5.2 ARID Concepts". The failure logic model was shown in Figure 3.56. The system model was also introduced as part of the modeling basics discussions in "3.5.1 Model Basics".

Figure 3.64 shows the system and its failure views again as a collage.

Example Analysis

The example analysis obviously starts with the assessment of the single service failures. In the example we assume that:



Assessing

Hazard requirement: 0.076

Select max situation: E3

Services

Service
s1
s2
s3
s4

Dependencies

Service

Figure 3.63: ARID's assessment dialog

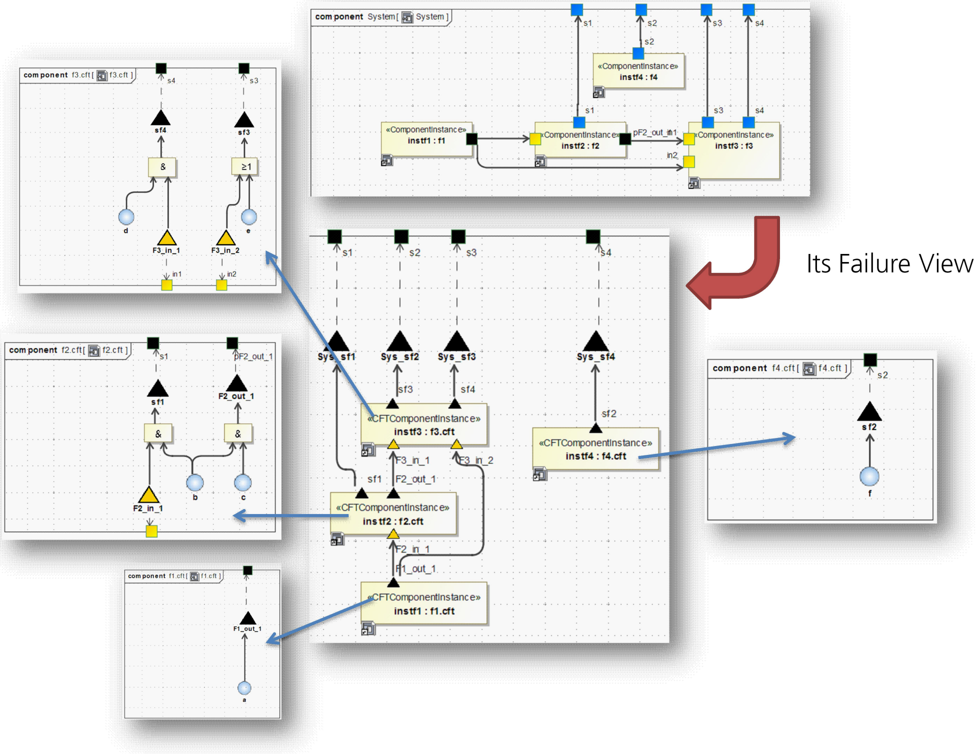
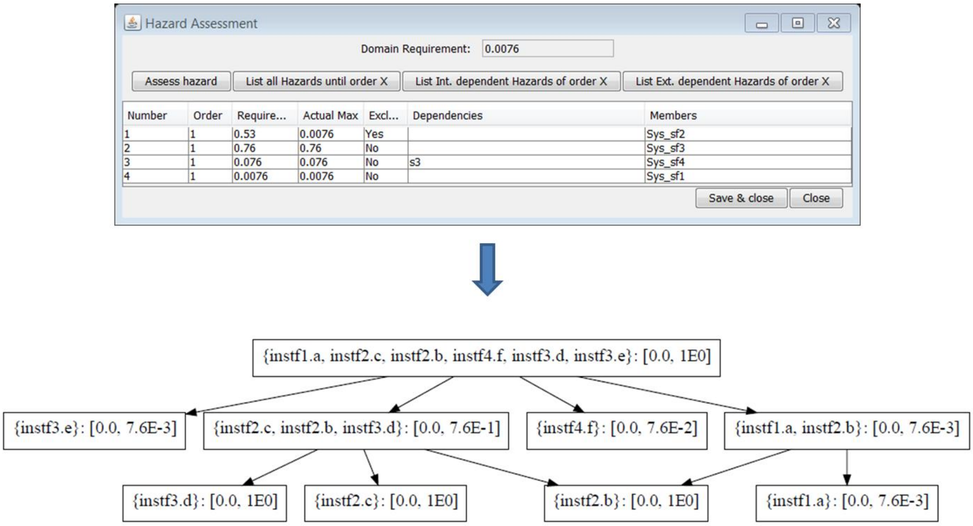


Figure 3.64: ARID's evaluation example

- The internally independent service failure "Sys_sf4" externally depends on service "s3" and has an ASIL C requirement.
- Service failure "Sys_sf3" has an ASIL B requirement.
- Service failure "Sys_sf1" has an ASIL D requirement.
- Service failure "Sys_sf2" has an ASIL A requirement.

Figure 3.65 shows as a result the assessment GUI and the prime implicants knowledge base (as a dotty graph). One interesting aspect is that the assessment of "Sys_sf2" was not necessary (did not result in any assessment knowledge): The actual requirement is much lower than the assessment result. The actual requirement was derived or is implied by the assessment of "Sys_sf1".
Thus, even though we did not design our method to improve the traditional case, it might happen that the domain knowledge can exclude hazards even in the very first step.



Domain Requirement: 0.0076

Assess hazard
List all Hazards until order X
List Int. dependent Hazards of order X
List Ext. dependent Hazards of order X

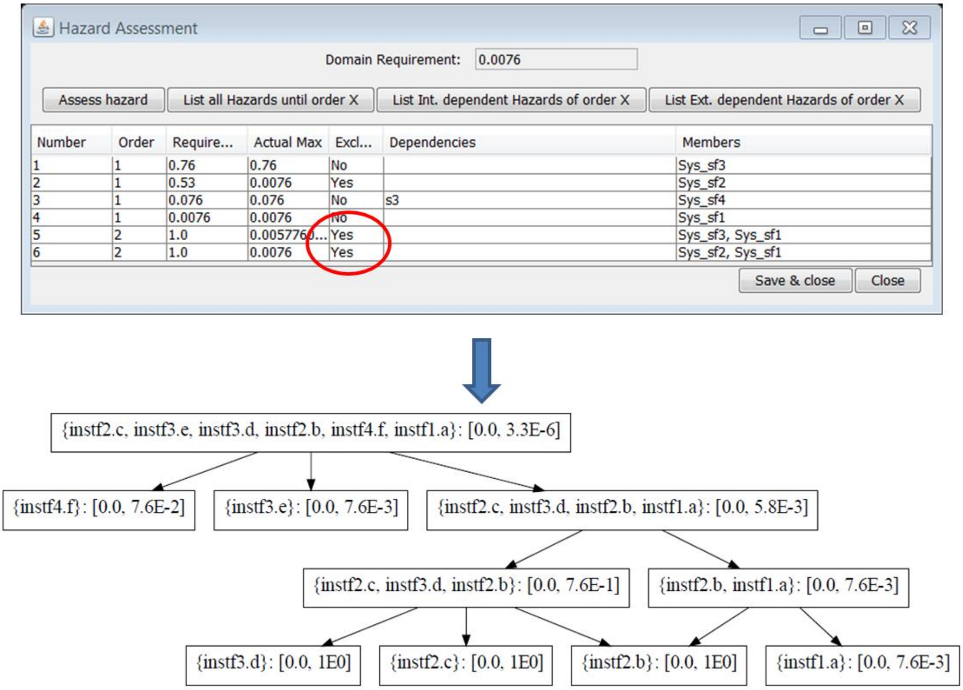
Number	Order	Require...	Actual Max	Excl...	Dependencies	Members
1	1	0.53	0.0076	Yes		Sys_sf2
2	1	0.76	0.76	No		Sys_sf3
3	1	0.076	0.076	No	s3	Sys_sf4
4	1	0.0076	0.0076	No		Sys_sf1

Save & close
Close
{instf1.a, instf2.c, instf2.b, instf4.f, instf3.d, instf3.e}: [0.0, 1E0]
{instf3.e}: [0.0, 7.6E-3]
{instf2.c, instf2.b, instf3.d}: [0.0, 7.6E-1]
{instf4.f}: [0.0, 7.6E-2]
{instf1.a, instf2.b}: [0.0, 7.6E-3]
{instf3.d}: [0.0, 1E0]
{instf2.c}: [0.0, 1E0]
{instf2.b}: [0.0, 1E0]
{instf1.a}: [0.0, 7.6E-3]

Figure 3.65: Step 1 of ARID's evaluation example

The next step in ARID's analysis process is to analyze all internally dependent hazards. In this example, only "Sys_sf3; Sys_sf1" and "Sys_sf2; Sys_sf1" are internally dependent. The automated analysis, however, shows that even though these hazards are internally dependent, they already fulfill the highest possible requirement. Figure 3.66 shows the (fully automatic) result of step 2.

The next step is to analyze all externally dependent service failures. In step 1, we defined only one dependent service failure. The algorithm yields that we need to assess the hazard; we do so without any further controllability

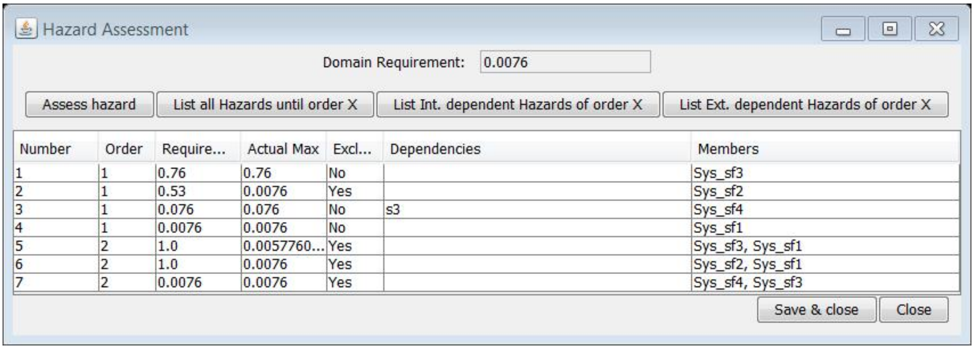

Hazard Assessment
Domain Requirement: 0.0076
Assess hazard
List all Hazards until order X
List Int. dependent Hazards of order X
List Ext. dependent Hazards of order X
Save & close
Close

Number	Order	Require...	Actual Max	Excl...	Dependencies	Members
1	1	0.76	0.76	No		Sys_sf3
2	1	0.53	0.0076	Yes		Sys_sf2
3	1	0.076	0.076	No	s3	Sys_sf4
4	1	0.0076	0.0076	No		Sys_sf1
5	2	1.0	0.0057760...	Yes		Sys_sf3, Sys_sf1
6	2	1.0	0.0076	Yes		Sys_sf2, Sys_sf1

{instf2.c, instf3.e, instf3.d, instf2.b, instf4.f, instf1.a}: [0.0, 3.3E-6]
{instf4.f}: [0.0, 7.6E-2]
{instf3.e}: [0.0, 7.6E-3]
{instf2.c, instf3.d, instf2.b, instf1.a}: [0.0, 5.8E-3]
{instf2.c, instf3.d, instf2.b}: [0.0, 7.6E-1]
{instf2.b, instf1.a}: [0.0, 7.6E-3]
{instf3.d}: [0.0, 1E0]
{instf2.c}: [0.0, 1E0]
{instf2.b}: [0.0, 1E0]
{instf1.a}: [0.0, 7.6E-3]

Figure 3.66: Step 2 of ARID's evaluation example

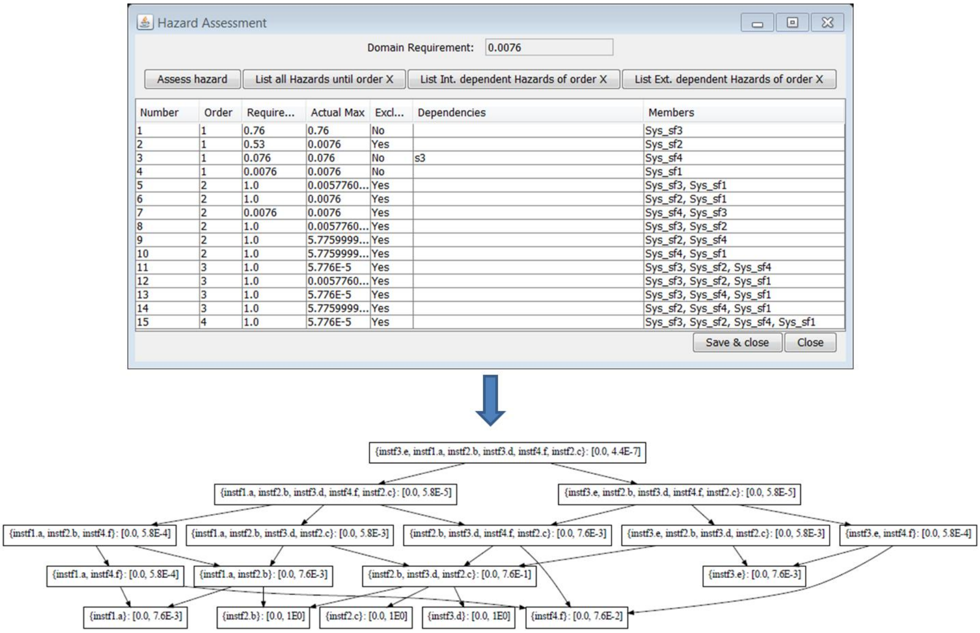
argument. The result is shown in Figure 3.67. (We omit the knowledge base in this figure.)


Hazard Assessment
Domain Requirement: 0.0076
Assess hazard
List all Hazards until order X
List Int. dependent Hazards of order X
List Ext. dependent Hazards of order X
Save & close
Close

Number	Order	Require...	Actual Max	Excl...	Dependencies	Members
1	1	0.76	0.76	No		Sys_sf3
2	1	0.53	0.0076	Yes		Sys_sf2
3	1	0.076	0.076	No	s3	Sys_sf4
4	1	0.0076	0.0076	No		Sys_sf1
5	2	1.0	0.0057760...	Yes		Sys_sf3, Sys_sf1
6	2	1.0	0.0076	Yes		Sys_sf2, Sys_sf1
7	2	0.0076	0.0076	Yes		Sys_sf4, Sys_sf3

Figure 3.67: Step 3 of ARID's evaluation example

The final step is to analyze all independent hazards. Due to the fact that we have four single service failures, the set of possible hazards has a cardinality of $2^4 - 1 = 15$. Currently, we have roughly half of the hazards analyzed (not manually assessed). The final (fully automatic) reasoning again results in the fact that all hazards already fulfill the domain requirement boundary and do not need to be assessed! Figure 3.68 shows the final assessment control center and the respective (final) knowledge base.


Hazard Assessment
Assess hazard
List all Hazards until order X
List Int. dependent Hazards of order X
List Ext. dependent Hazards of order X
Save & close
Close
{instf3.e, instf1.a, instf2.b, instf3.d, instf4.f, instf2.c}: [0.0, 4.4E-7]
{instf1.a, instf2.b, instf3.d, instf4.f, instf2.c}: [0.0, 5.8E-5]
{instf3.e, instf2.b, instf3.d, instf4.f, instf2.c}: [0.0, 5.8E-5]
{instf1.a, instf2.b, instf4.f}: [0.0, 5.8E-4]
{instf1.a, instf2.b, instf3.d, instf2.c}: [0.0, 5.8E-3]
{instf2.b, instf3.d, instf4.f, instf2.c}: [0.0, 7.6E-3]
{instf3.e, instf2.b, instf3.d, instf2.c}: [0.0, 5.8E-3]
{instf3.e, instf4.f}: [0.0, 5.8E-4]
{instf1.a, instf4.f}: [0.0, 5.8E-4]
{instf1.a, instf2.b}: [0.0, 7.6E-3]
{instf2.b, instf3.d, instf2.c}: [0.0, 7.6E-1]
{instf3.e}: [0.0, 7.6E-3]
{instf1.a}: [0.0, 7.6E-3]
{instf2.b}: [0.0, 1E0]
{instf2.c}: [0.0, 1E0]
{instf3.d}: [0.0, 1E0]
{instf4.f}: [0.0, 7.6E-2]

Domain Requirement: 0.0076

Number	Order	Require...	Actual Max	Excl...	Dependencies	Members
1	1	0.76	0.76	No		Sys_sf3
2	1	0.53	0.0076	Yes		Sys_sf2
3	1	0.076	0.076	No	s3	Sys_sf4
4	1	0.0076	0.0076	No		Sys_sf1
5	2	1.0	0.0057760...	Yes		Sys_sf3, Sys_sf1
6	2	1.0	0.0076	Yes		Sys_sf2, Sys_sf1
7	2	0.0076	0.0076	Yes		Sys_sf4, Sys_sf3
8	2	1.0	0.0057760...	Yes		Sys_sf3, Sys_sf2
9	2	1.0	5.7759999...	Yes		Sys_sf2, Sys_sf4
10	2	1.0	5.7759999...	Yes		Sys_sf4, Sys_sf1
11	3	1.0	5.776E-5	Yes		Sys_sf3, Sys_sf2, Sys_sf4
12	3	1.0	0.0057760...	Yes		Sys_sf3, Sys_sf2, Sys_sf1
13	3	1.0	5.776E-5	Yes		Sys_sf3, Sys_sf4, Sys_sf1
14	3	1.0	5.7759999...	Yes		Sys_sf2, Sys_sf4, Sys_sf1
15	4	1.0	5.776E-5	Yes		Sys_sf3, Sys_sf2, Sys_sf4, Sys_sf1

Figure 3.68: Step 4 of ARID's evaluation example

3.5.5 Conclusion of ARID Analysis Example

In the example shown, we only had to manually analyze five out of a total of 15 possible hazards. For all other hazards we can assure that the risk is below the maximally acceptable boundary. Compared with the traditional analysis, which would perform four manual assessments, we have created much more confidence by just adding one more assessment (and applying ARID's reasoning).

The performance and the possibilities for hazard exclusions are, of course, strongly dependent on the assessment results and the "strength" of the derivable knowledge. This example was not meant to show efficiency, but feasibility and proof of concept. Due to the fact that this example is synthetic and the assessment results were selected by the authors, we performed a more realistic case study within the "e performance" project.

The second case study was applied to the system model of an electrical power train. The system comprised five services and eight service failures. The complete case study can be read in the thesis of Pierre Iraguha [Ira11].

Table 3.9: The result statistics as an overview

Criterion	value
Single service failures	8
Number of all possible hazards	255
Number of manual assessments	
Traditionally:	8
With ARID:	13
Number of assessments per step	
Step 1:	8
Step 2:	4
Step 3:	1
Step 4:	0
Percentage of overall hazards	
Traditionally:	$\frac{8}{255} = 3.1\%$
With ARID:	$\frac{13}{255} = 5.1\%$
Coverage of all possible hazards	
Traditionally:	$\frac{8}{255} = 3.1\%$
With ARID:	$\frac{255}{255} = 100\%$
Additional effort ARID produces	
Absolute:	5
Relative:	2%
Increase in confidence	
Absolute:	96.9%
Relative:	$\frac{96.9\%}{3.1\%} = 3126\%$

Some statistics about the case study are shown in Table 3.9. One can see that in the first step, all single service failures were assessed. Due to the criticality of the system, the assessment results yielded high ASIL requirements and subsequently it was possible to omit many multiple service failures from the assessment. This becomes clear if we investigate the necessary manual assessments in steps 2 - 4. While in step 2, "4" additional analyzes were necessary, step 3 required only one additional analysis (although we had three controllability arguments). Finally, in step 4, no manual assessment was necessary! Thus the increase of manual assessment compared to the traditional assessment was $5/8 = 62\%$. On the one hand, this is additional effort, but on the other hand we have automatically analyzed $2^8 - 1 - 13 =$

242 additional hazards. Thus, compared to the overall number of all possible hazards, we assessed only $\frac{13}{255} = 5.1\%$ of all hazards the system is subject to. (The traditional approach would analyze $\frac{8}{255} = 3.1\%$.)

Because one has to spend additional effort, the question arises of whether this effort should be spent or not. One aspect is, of course, that ARID delivers 100% confidence for the safety engineer that he did not miss anything. However, a closely related question is if any safety requirements are missed if only single service failures are assessed. One hypothesis is that ARID does not deliver any additional safety requirement. We proved this hypothesis wrong using our industrial case study and showing that the traditional approach misses essential safety requirements.

Safety requirements are essentially given by different boundaries for basic events or basic event combinations. We therefore compared the quantitative requirements of the basic events after the first step, the traditional analysis, with the final requirement after ARID: We derived that eight basic events (out of 25) had insufficient safety requirements. To seven of them, the traditional analysis assigned the lowest level of QM and to one an ASIL A. This means that only one of them would have had any safety requirement assigned. ARID's result was that all of them had the highest level assigned to them - ASIL D. Thus, in reality these failures are highly critical, but no safety engineer would have paid any attention to seven of them and to one of them only with minor rigor. The reason for this was that all of them contribute to multiple service failures, but each single service failure by itself has only QM or ASIL A "criticality". Even though this is only one example, we claim to have refuted the hypothesis that the effort is not worth being spent.

4 Evaluation

In the previous chapter, we presented our solution approach SAHARA and its building blocks GOBI, OASIS, HEAT, and ARID. As part of the single blocks, we already discussed evaluations and feasibility studies.

In this chapter, we will not repeat all aspects of our evaluation(s) but concisely summarize our evaluation results and thereby argue our contribution again. For details of the respective evaluations, we will refer to the respective section.

We will apply the following recurring argumentation structure to each building block:

1. We will state the problem addressed by the building block as discussed in the problem statement section 1.2.
2. We will repeat our contribution of the stated problem, as presented in section 3.1.
3. We will present our evaluation summary showing why we think our solution actually solves the problem.

As discussed in the problem statement, we are aiming at a holistic, model-based approach to a structured HRA process. The first building block, GOBI, served the purpose of providing an integration framework. Therefore, GOBI as a conceptual model, constitutes the prerequisite for our core contributions OASIS, HEAT, and ARID. We did not explicitly evaluate the validity of the GOBI model, but assume its validity implicitly by showing that our core contributions could be evaluated successfully.

In this section, we will present the evaluation of our core contributions and discuss their validity. Following our overall breakdown structure of the problem statement, we evaluated our contributions with regard to the two challenges identified in the problem statement:

- Consistency and reuse of operational situations (this is addressed in OASIS and HEAT)
- Coping with hazards consisting of multiple service failures (this is addressed in ARID)

Figure 4.1 depicts the problems on the left, the solution block(s) we evaluated in the middle and the claimed advancement of the state of the art on the right.

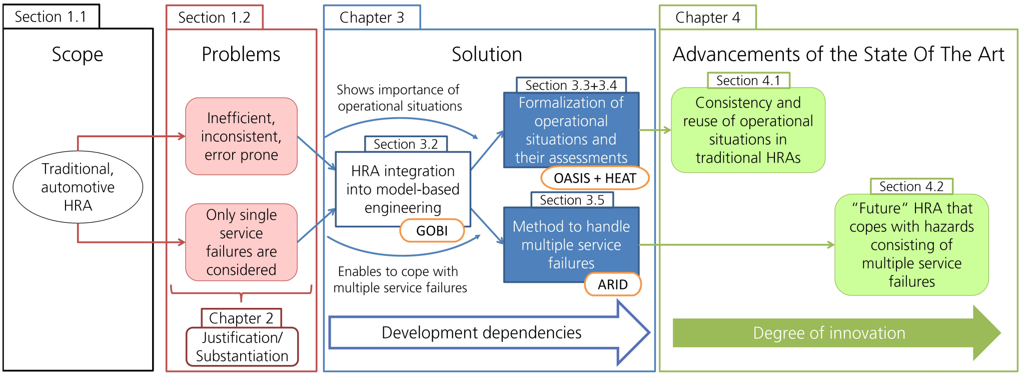

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error prone
Only single service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Enables to cope with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method to handle multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State Of The Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 4.1: Overview over evaluation structure

4.1 Evaluation of OASIS and HEAT

As discussed in section "Problem Statement", the challenge of formalizing operational situations and their assessments was broken down to the technical challenges as follows:

- ✧ Structuring of risk assessments
  - ✧ Correctness of situation analysis
    - ➠ Formalization of operational situations
      ↪ OASIS - Ontology-based Analysis of Situation Influences on Safety
    - ➠ Consistency and reuse of operational situation assessments
      ↪ HEAT - Handy Exposure Assessment Technique

In the discussion of the problem statement, we identified the challenge of situation analysis correctness. The core aspects of this challenge were the consistency of situation assessments and their reuse. In order to achieve this, one fundamental prerequisite was the formalization of operational situations. In OASIS we addressed this prerequisite, and with the evaluation we answered the question of whether it is possible to formalize HRA situations. In HEAT we rely on the OASIS' formalization of operational situations, then add consistency checks and provide a reuse approach for operational situation assessments.

4.1.1 Evaluation of OASIS

For OASIS we evaluated the validity of the claimed contribution:
We developed an ontology-based situation influence factor classification by classifying influence factors and integrating HRA-relevant domain knowledge into an ontology. This allowed us to propose a structured situation definition approach.

The formalization of operational situations finally referred to two main aspects: (1) structuring possible influence factors of an operational situation and (2) providing a process for using these influence factors in a way that allows coming up with actual situations. Therefore, our evaluation had two aspects as well. We proved influence factor structuring by formalizing the influence factors of nine industrial case studies, the examples given in ISO 26262, and examples we developed on our own. The overall number of situation descriptions was around 300 and the overall number of influence factors extracted and formalized was around 200. Each influence factor was classified into one of the OASIS classes. Figure 3.14 on page 71 shows the distribution between the first two levels of the identified relevant classification concepts. We found that using the hierarchy of classes OASIS provides, it was very straightforward to assign a class to each influence factor. One example of an influence factor to classify is "foliage": From the top level it is clear that it is an "Environmental Characteristic" (not something belonging to the human or the vehicle). The second level of classes are "natural condition" and "usage location". In OASIS, these classes are separated roughly by being "purely natural" or "man-made". "Foliage" is obviously purely natural and therefore belongs to the class "natural condition". The next refined class is "ground condition" and, even more specifically it is "natural coverage". One can follow this classification breakdown by looking at Figure 3.11 on page 66. Thus, the classification significantly helped in classifying the influence factors. Another result of the evaluation was showing feasibility: It was possible to integrate all influence factors into one of these classes.

The second part of the evaluation is given by the example "Powersteering" (cf. end of 3.3.4). For this (industrial case study) we showed how to use our process and the captured domain knowledge (from the former step) and support the process of complete situation definitions. We evaluated completeness in the sense that all situations that can be built using the influence factors contained in the OASIS ontology are either defined or are (provably) irrelevant. One important aspect is how much manual effort the definition of the complete set of relevant operational situations costs. In our case study, nine hazards were analyzed. In the first step, all usage locations (a total of 14) had to be analyzed manually. For the next step, the OASIS knowledge base delivered (automatically) pre-selected restrictions on possibly relevant combinations of (1) hazards and usage locations and (2) hazards and maneuvers. The number of possible combinations in case (1) - hazards and usage locations - was restricted to 85 reasonable combinations out of a total of 138. Thus, roughly 60% remain. With the physical constraints implied by the hazards and the physical model contained in OASIS, the relevant combinations of (2) - the relevant maneuvers per hazard - were automatically reduced from 138 to 106. One example of this automatic reduction is: The hazard "delayed steering" physically implies that one wants to steer, i.e., that the steering angle is not "0". The domain knowledge contained in OASIS provides the information that "driving straight" implies a steering angle of "0". Therefore, this combination can be computed as being not relevant. The restriction of these two combinations does not mean, however, that one needs to manually assess 85 and 106 combinations. Similar to the extraction of relevant information, we can use OASIS knowledge to combine these two by reasoning on which combinations of maneuvers and usage locations make sense. Thereby we integrate the aspects of hazard, maneuver, and usage location. This allowed further reduction of the number to 52 combinations out of a total of 756 possible combinations. These 52 combinations were assessed manually. The result of the final (manual) assessment was that all 52 combinations are relevant; for all other combinations, OASIS delivered an explanation as to why the combination is not relevant. Thus, the manual effort was reduced from 1890 to 14 * 9 + 52 = 178. (1890 is the product of 9 hazards, 14 usage locations, and 15 maneuvers). Thus, with OASIS, less than 10% of the effort was necessary compared to a traditional analysis.

Evaluation result of OASIS:
The evaluation has proven two things:
(1) It is possible to formalize a situation by using our proposed classification

of influence factors: the OASIS ontology.
(2) Our proposed approach for defining situations results in a complete set of situations with less effort: in our case study, less than 10% compared to the traditional analysis. However, the approach is only complete with respect to situations that can be built using influence factors contained in the OASIS ontology.

Limitations of OASIS:
One limitation of our situation definition process is that it relies on the completeness of the influence factors in the ontology (for each class used). Thus, one could redefine our achieved completeness as being complete with respect to already formalized domain knowledge.

4.1.2 Evaluation of HEAT

The existence of HEAT can be interpreted as another proof for the validity of the situation formalization OASIS delivers. Without formalized operational situations, it would not be possible to check consistency and provide a reuse approach.

In the evaluation of HEAT, we proved the validity of our contribution:
Situation modeling and exposure assessment support, through model-based situation representation with a similarity function, for reusing assessment results of (similar) situations to support the safety engineer in consistent situation assessments.

The first claim is that we can automatically check the consistency of a set of situation assessments. To evaluate this, we filled our situation knowledge base (which is provided by OASIS) with data from 9 industrial case studies, which led to 148 situations (including their industrially determined exposure value). As discussed in section 3.4.3, the evaluation was very successful. We could (fully automatically) identify 19 exposure inconsistencies among these 148 situations; thus, a factor of roughly 13% inconsistent situations. This was already a huge benefit for the industrial partner. This is not only true because we found inconsistencies, but the industrial case studies had been manually checked for consistency before and we could prove that a manual consistency check is insufficient. In addition to the deficient result, the effort for the manual checking took four experts about two weeks of effort. Our automated analysis took around five minutes of calculation time. One disadvantage was that the formalization as HEAT situations took around twice as long as the descriptive notation in an Excel table. In order to check whether this is a drawback of the method or the result of insufficient tooling support, we double-checked the elicitation of the formalized information by typing exactly the same information into a (structured) Excel template. In this case, no significant increase in effort was observable. Thus, by improving the tooling, we could remove this drawback to improve practical applicability.

The second and main aspect of the problem statement was to enable efficient reuse of already assessed operational situations. During our evaluation, we answered two questions:
(1) Is there potential for reusing situations?
(2) Does the approach support the safety engineer in performing a consistent assessment?
To evaluate these questions, we reused our nine industrial case studies. The finding for the first question was that we identified 14 duplicated situations. We considered these as candidates that could have been found using existing approaches, such as Excel tables: One can easily search for a situation in an Excel table and reuse the example. Thus, we asked ourselves in how many cases our approach would have supported the engineer by presenting similar situations. The indicator we used for this is the hierarchy of similar operational situations HEAT calculates. The hierarchy describes a generalization/specialization relation of operational situations. A more general situation restricts the upper bound of possible exposure values, whereas more special operational situation restricts the lower bound. Thus, if there is a hierarchy, this means the situations are similar and they are candidates the approach would propose as similar situations with their exposure values as (consistent) exposure boundaries. The number of situations contained in a hierarchy was 60: $44 + 12 + 4 = 60$ - cf. Table 3.3 "Partial-order chain lengths" on page 127. Thus, in almost 40% of the cases HEAT would have supported the situation assessment compared to the 9% maximum of possible reuse in traditional HRAs.
The other question, whether HEAT would help to produce a *consistent* assessment, was not really a question anymore. Due to the construction that HEAT proposes only consistent assessment values, it was obvious that consistency with the knowledge base is not a problem anymore - the safety engineer simply has to select one of the proposed (consistent) exposure values.
This strong consistency, ensured via HEAT's assessment support, can, however, be a drawback as well. Due to the fact that the likelihoods of some influence factors are based on statistics and that sometimes different statistics arrive at different likelihood values, it is possible that one situation has different assessment results. In the current state of the knowledge base, this would lead to an inconsistency and the knowledge base would not be usable anymore. However, this limitation is only given for the current implementation of the OASIS ontology. As described in HEAT, one can easily extend the OASIS ontology by other separating facts. One solution of this limitation would be, for example, adding the respective statistic as one parameter of a situation. This would make the situation only similar and not identical anymore, and our knowledge base could be used again.

Evaluation Result of HEAT:
The evaluation has shown that it is possible to check the consistency of a knowledge base. In our evaluation, the consistency check was not only better but, compared to the manual consistency check, almost instantaneous. Another aspect we evaluated was whether there is a potential to

reuse situations. In our case study, 10% of the situations could have been reused in existing approaches. In contrast, HEAT would have supported the assessment in 40% of the cases by presenting similar situations.

Limitation:
We see limitations that stem from the current implementation and tools:
The first limitation is that modeling situations in our tool is laborious. As stated above, this is not a general limitation of the method, but a result of missing tooling support.
Another limitation is that HEAT requires consistent situation assessments within the knowledge base, even though it is, in practice, possible to have different assessment results for one situation. However, this is not a drawback of our method, but rather a matter of using the extension mechanism of our OASIS ontology to resolve the inconsistency. We did not implement this because this is a subtlety for practical application and not a matter of scientific contribution. Additionally, for practical applications, we experienced two standpoints: Some industrial partners want to have these different statistics and would need a way to handle this, while others consider exactly the information that there are inconsistencies due to different statistics as an advantage and propose coming up with one statistic or value that is considered to be valid in the automotive domain.

4.2 Evaluation of ARID

As discussed in section "Problem Statement", the challenge of coping with multiple service failures was structured as follows:

- ✧ Structuring of hazard analysis
  - ✧ Completeness of hazard analysis
    - ✓ Identification of system-level failures
    - ➠ Dealing with multiple service failures

The overall challenge "Completeness of hazard analysis" focuses on completeness of the analysis, i.e., on whether all possible hazards were analyzed or not. Due to the fact that a hazard is a set of system-level failures, one needs to identify the system-level failures as a prerequisite for the hazard analysis. As discussed in the related work, the identification of system-level failures is no scientific challenge anymore. Therefore, in ARID we focus on completeness of the analysis by dealing with hazards that consist of multiple service failures.

Our claimed contribution is:
Approach for dealing with multiple service failures. This includes a model-based representation of multiple service failures and a corresponding method defining a computer-aided analysis process.

The detailed evaluation was presented in section 3.5. In this section, we summarize the findings of the evaluation and answer two questions:

(1) Does the analysis of hazards consisting of multiple service failures yield any new information or, does the traditional approach miss safety-relevant requirements?
(2) Is it possible to analyze *all* multiple service failures with ARID?

To answer these questions, we performed two case studies, one small case study for which *we* provided the model and another industrial case study. Because the latter originated from industry, it represents a realistic scenario for the application of ARID.

We evaluated the first question using the industrial case study. The case study comprised almost the complete power train of an electrically driven car with wheel hub motors. In the end, we compared the safety requirements that were determined using a traditional HRA and those determined using ARID. The failure model of that case study contains 25 basic events, meaning 25 possible elements of the power train that could fail. With ARID, we derived that eight basic events had insufficient safety requirements. The traditional analysis assigned the lowest level of QM to seven of them and an ASIL A to one. This means that only one of them would have had any safety requirement assigned. After performing an ARID analysis, all of them had the highest level assigned to them - ASIL D. Thus, in reality these failures are highly critical, but to seven of them, no safety engineer would have paid any attention, and to one of them only with minor rigor. The reason for this was that all of them were analyzed as single service failures. One concrete case identified was the following: If one analyzes that one cannot accelerate anymore, this is not too hazardous because one can steer to the side and turn on the warning lights. If the boost of the power steering fails, one expects the driver to be able to steer the car without any support, thus, this hazard is not critical. If the stability control is lost, this is not critical in most cases because the driver should be able to control the car on his own by using the brakes, the acceleration, and the steering capabilities. However, in the case study one identified basic event led to a multiple service failure because the electrical power system was affected. The basic event described the fault that no energy is available anymore and the engines are not shut down. The shutdown of the engines was important in the case study because the electrical engines are blocked if they are not energized and not short-circuited. The result on the system level is that the driver cannot accelerate, the boost of the power steering is lost, and additionally stability is lost because the engines are blocked. In this case of multiple service failures, no driver can handle the situation anymore, i.e., it is highly critical. The reason why this case was not considered was that the basic event got multiple times a QM and ASIL A assigned. After the assignment of ASIL A, each additional single service failure that resulted in an ASIL A (or QM) was assumed to be covered already (because the basic event already had ASIL A assigned to it). The case that a multiple assignment of ASIL A could result in a higher ASIL, in this case ASIL D, was not considered because no multiple service failure analysis was performed. While this is only one case study, we assume that such cases are the rule, not an exception. We therefore state that it is not acceptable to only perform hazard analyses for single service failures. We furthermore strongly vote for rising the note in ISO 26262 saying that multi-functional degradation should be analyzed to an explicit and exposed normative requirement.

The other question we addressed in the ARID evaluation is whether it is feasible/efficient to analyze all possible hazards. For this we performed two case studies. We used the smaller, theoretical case study to check whether ARID's algorithm works as expected. We did this by explicitly investigating the progress and the changes in ARID's internal knowledge base. We showed that the concepts and algorithms work as expected and that the knowledge base was produced as expected. The final result was that we had to manually assess five out of maximum of 15 hazards. This is one additional manual assessment compared to the traditional approach, but we can guarantee that all possible hazards have been analyzed, including those consisting of multiple service failures. Due to the fact that the system was theoretical and that it was built by us for the purpose of evaluation, these numbers do not seem to deliver hard proof.
We therefore performed another larger (industrial) case study. The system and the failure propagation of the case study model were not originally built for ARID analyses and are therefore representative. The results of this case study were discussed in section 3.5.5. The main concern when moving from single service failures to multiple service failures was the exponential number of assessments or analyses that need to be performed. However, this is only a problem if these need to be analyzed manually. If they are analyzed automatically, it is not a problem. In our case study, only 13 manual assessments had to be performed with ARID, whereas the traditional approach required eight manual assessments. Thus, the manual assessments increased by 5; the assessments performed automatically are 242 with ARID and 0 traditionally. Thus, with the traditional approach, we analyzed only 3.1% (= 8) of all hazards. Using ARID we achieved a coverage of 100% (= 255) and had to assess only 5.1% (= 13) manually. Thus, with only minor additional (manual) effort, we were able to analyze all possible hazards. In addition to this increase in hazard analysis coverage, we found insufficiencies with respect to safety requirements if only the traditional approach had been applied (as discussed above).

Evaluation result:
The evaluation has shown that with our method it is possible to practically analyze all hazards of a system, while the traditional approach only analyzes single service failures. Thus, using ARID, multiple service failures can be handled. Additionally, we uncovered that, at least for our industrial case study, the traditional approach results in insufficient safety requirements.

Limitation:
ARID strongly relies on the availability of system models and corresponding failure propagation models. This is currently a limitation of ARID because we experienced that these models are rarely found in practice.

4.3 Overall Evaluation Result

The fundamental challenge we addressed in this thesis was how to structure and formalize HRAs in order to make the result less dependent on experts' experience and to increase the consistency of the result of an HRA.

As the name "Hazard analysis and risk assessment" implies, the overall task can be subdivided into two steps: (1) hazard analysis and (2) risk assessment. The result produced in the first step, the hazard analysis, are lists of hazardous events. A hazardous event is a combination of a hazard and an operational situation, which needs to be assessed in the second step. The fundamental challenge of being more consistent and less dependent on experts' experience for this step essentially means formalizing the description of operational situations. With OASIS, we provided a methodology that allows the safety engineer to define situations more formally. We evaluated this using several industrial case studies and were able to show the effectiveness of OASIS for defining operational situations. Thus, with the formalization of operational situations, we advanced the state of the art for defining hazardous events.

The next step (2) is the risk assessment. In this step, the expert needs to come up with an integrity level reflecting the necessary risk reduction that needs to be achieved during the development. In the automotive domain, the determination of the integrity level is based on determining three parameters: exposure, controllability, and severity. To achieve the fundamental goal of making the HRA less experience-dependent, the determination of these three parameters needs to be more structured and less expert-dependent. The input for the first parameter is only the respective operational situation, while the second and third parameters are determined by the combination of the hazard and the operational situation. Due to the importance of the operational situations we focused on formalizing the Exposure parameter. The formalization of the determination of the two parameters Controllability and Severity is, of course, important as well, but due to the fact that formalizing Controllability requires a lot of knowledge about human behavior and human capability models, this aspect was beyond the scope of this thesis. The same is true for the "Severity" parameter: One would need a very good understanding of the criticality of human injuries, which we cannot provide in this thesis.

Due to the fact that the exposure is the likelihood of an operational situation (without the hazard), we tightly coupled the assessment of the likelihood of the operational situation, addressed in HEAT, with its description, addressed in OASIS. In our evaluation, we were able to show that checking consistency among a set of assessed situations requires not only orders of magnitude less effort, but also that we achieved a higher degree of consistency.

With these two steps we were able to achieve a more structured approach for HRAs than what is performed in modern industrial practice. From a scientific point of view, another important issue is the consideration of multiple service failures as hazards. If one considers multiple service failures as relevant hazards, the challenge arises that the number of hazards grows exponentially with the number of single service failures. Using a traditional approach, it is not possible to cope with this complexity. We therefore extended our model-based HRA approach by ARID and could show that it is feasible to analyze the exponential number of hazards. This is possible because ARID makes extensive use of model-based information to automatically reason whether a hazard needs to be assessed manually or not. Another finding of our evaluation of ARID was that the traditional approach of only considering single service failures leads to insufficient safety requirements.

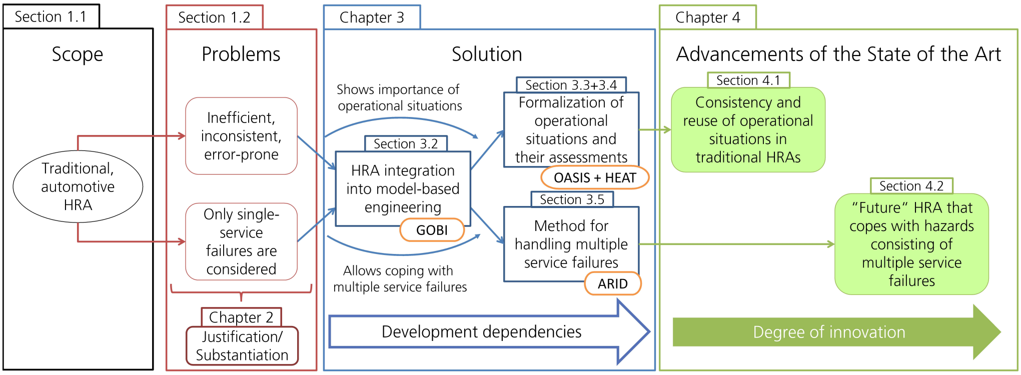

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error-prone
Only single-service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Allows coping with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method for handling multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State of the Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 4.2: Our advancements of the state of the art

Figure 4.2 shows our two steps of advancements of the state of the art: The first step made the HRA as it is traditionally performed more consistent, more efficient, and less dependent on experts' experience. The most important aspect for this was the formalization of operational situations and their assessments. Based on this more formal, model-based representation of HRAs we advanced the state of the art even further by providing a method that allows coping with multiple service failures. We additionally proved that this is, in practice, a necessary step by showing that by using ARID, stricter safety requirements are created than in the traditional analysis.

The limitations of our approach are that we did not formalize the assessment of controllability and severity. This has to be done in a traditional way and there is neither a way to check the consistent assessment of these parameters nor is it possible to efficiently reuse the result of these assessments. If this were to be done, the overall result would probably be even more consistent and efficient. We omitted these parameters because the operational situation is more important on the one hand, and because on the other hand, the knowledge needed for the formalization is out of scope: One needs a good background in human capability modeling for controllability assessments and in medicine for formalizing the severity.
The fundamental limitation of our approach to handling multiple service

failures is that we need appropriate system and failure propagation models in order to perform the analysis. However, these models are seldom available in modern automotive practice.

5 Summary and Conclusion

In this section, we will first give a short summary of the contributions of SAHARA with respect to advancements of the state of the art. In the second subsection, we will reflect on the overall work, the challenges, and the lessons learned during our work on SAHARA. We conclude this section with a short outlook on future work.

5.1 Contributions

In this thesis, we advanced the state of the art of hazard analysis and risk assessments (HRA) in the automotive domain. The HRA is a very important step for the safety of automotive systems because in this step, the safety goals are determined. The subsequent task of safety engineering is to ensure the correct implementation of these safety goals.

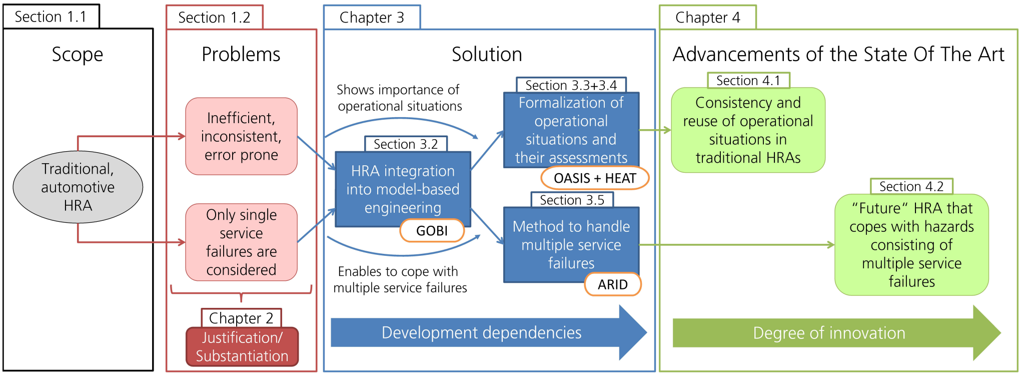

Section 1.1
Scope
Traditional, automotive HRA
Section 1.2
Problems
Inefficient, inconsistent, error prone
Only single service failures are considered
Chapter 2
Justification/ Substantiation
Chapter 3
Solution
Shows importance of operational situations
Section 3.2
HRA integration into model-based engineering
GOBI
Enables to cope with multiple service failures
Section 3.3+3.4
Formalization of operational situations and their assessments
OASIS + HEAT
Section 3.5
Method to handle multiple service failures
ARID
Development dependencies
Chapter 4
Advancements of the State Of The Art
Section 4.1
Consistency and reuse of operational situations in traditional HRAs
Section 4.2
"Future" HRA that copes with hazards consisting of multiple service failures
Degree of innovation

Figure 5.1: Schemantic overview of SAHARA

Although this step is highly important, the current state of the art of HRAs lacks efficiency, consistency and correctness. The quality of the result is to a large extent dependent on experts' experience. The most important aspect, which is currently handled very informally, is the definition of operational situations that are relevant when a hazard occurs. (Please recall that the determination of hazards is not a challenge in the current state of the art.) Thus, one advancement is to formalize operational situations and combine this with the already existing approaches for hazard identifications. This makes the current process of HRAs more consistent, more efficient, and increases the correctness of the results. Besides improving the HRA as performed nowadays, another drawback of the current state of the art of HRAs is that only single service failures are considered as hazards. Due to the high integration of functions and the ever increasing functional dependencies, one has to consider multiple services failing at the same time as well. If one transitions from single service failures to multiple service failures, the big challenge is that this means that the number of hazards to be analyzed grows exponentially with the number of single service failures. With the current state of the art, this cannot be handled because currently every hazard needs to be analyzed manually. This means that a safety expert would have to perform an exponential number of hazard analyses.

For both challenges of the current state of the art, we pursued a model-based formalization of the HRA as our fundamental solution with the HRA embedded in a model-based engineering approach, it is easier to assure consistency and correctness of the results on the one hand; on the other hand, it enables computer-aided hazard analysis, which we consider the only way to handle an exponential number of hazards. Thus, our overall and fundamental first step was to translate the HRA into a model-based representation. As our basic modeling approach, we reused Parnas' Four-Variable-Model (FVM). However, we extended it to make the three most important aspects in an HRA more explicit: We created an interacting Four-Variable-Model, with the System, the Human, and the Environment being the single FVMs. The big advantage is that we can talk more explicitly about the operational environment and therefore about operational situations. The assessment of the likelihood of operational situations is termed exposure. This parameter, as well as the other two parameters controllability and severity, were discussed in the context of our interacting FVMs. The formalization of controllability and severity was beyond the scope of this thesis. However, we presented how our model can be extended, by sketching the integration of Rasmussen's Skill-Rule-Knowledge model to represent controllability actions more formally. We termed this formalization step GOBI. It serves as the formalization and model-based engineering basis for coping with the challenges stated above.

5.1.1 Consistency and reuse of operational situations in traditional HRAs

As discussed above, the first challenge was to make the HRA more consistent and more efficient and to increase the correctness of the results. We identified the most important aspect to be formalized: the operational situation. Operational situations in combination with hazards are the basis for the determination of the three parameters Exposure, Controllability, and Severity. The determination of exposure is done solely by investigating the operational situation (without the hazard). Because the definition of situations and the determination of their likelihood is separated in the traditional process separated, we divided our approach to situation definition and exposure likelihood determination into OASIS and HEAT, respectively. However, the results are highly cohesive contributions.

With OASIS, we aimed at making operational situations more formal in the sense that we structured and classified the building blocks of an operational situation, the influence factors. From the state-of-the-art review we learned that ontologies are suited best for representing and structuring information. We consolidated our view on ontologies especially in the differentiation to meta-models. Afterwards we developed an extensible classification of HRA-relevant influence factors as an ontology. For that we used the model-based HRA representation provided by GOBI and used our separation of "System", "Human", and "Environment" as the first-level classification. We discussed and derived the second level of classes using over 300 operational situations contained in 10 industrial case studies. We do not claim that the ontology is complete with respect to all possible influence factors, but we claim that the classification *structure* is complete. I.e., all influence factors found later on can be integrated into the defined structure. Furthermore, the influence factors are complete to represent the 300 operational situations mentioned above.

Besides the influence factors themselves, the dependency information with respect to their occurrence likelihood is important domain knowledge as well. If we consider, for example, the two influence factors "$v_x > 100km/h$" and "*city*", we can add the domain knowledge that this combination is highly unlikely. Another example would be the implicit knowledge that "$v_x > 100km/h$" implies that there is a "*driver*" operating the vehicle. We therefore extended our influence factor classification ontology to allow this domain-knowledge to be captured as well.

With these two steps we formalized the implicit knowledge structure a safety engineer has in mind and filled the ontology with data from ten case studies. We furthermore defined a situation definition process that uses our formalized information (stored in the ontology) to construct actual operational situations.

Once we had a more formal definition of operational situations (with OASIS), we extended this to include the assessment information, the exposure, of operational situations in our ontology. Our focused goal of consistency and efficiency became important in this step. Situations are consistent if their assigned exposure values are consistent. Efficiency with respect to operational situations is also obtained by reusing already existing assessment knowledge, i.e., the exposure value of operational situations. Thus, stored assessments constitute domain knowledge that can be used to support subsequent assessments. One can either retrieve former exposure values, or guide the safety engineer to a consistent assessment of a situation that is not contained in the current knowledge base. The case of exact retrieval (like simple database queries offer) is not difficult. It is more sophisticated and universal to work with *similar* situations:
One would say, for example, that "*driving straight on highway with* $100km/h \leq v_x < 130km/h$" is very similar to the situation "*driving straight on a country road with* $80km/h \leq v_x < 100km/h$", but is very different from "*parking car with children playing inside*".

We therefore constructed a function that uses the OASIS influence factor classification to calculate influence factor and operational situation similarity. This function was integrated into a case-based reasoning engine that works on ontologies. However, we had to adapt the case-based reasoning to work with our extended OASIS ontology on the one hand and with our newly defined situation similarity function (in the case retrieval stage), on the other hand.

We finally used the ontological CBR approach to support the safety engineer in two ways:
(1) To check the consistency of a set of assessed situations, and
(2) to support the safety engineer in assessing (new) situations consistently, by providing results of previous assessments.

Thus, with the combination of OASIS and HEAT we provided an approach to more formal operational situations and, therefore, more consistent assessment results of operational situations. In combination with our integrating HRA framework, GOBI, we therefore advanced the state-of-the-art approach of HRAs by making it more consistent, efficient, and less dependent on experts' experience.

5.1.2 HRA approach that copes with hazards consisting of multiple service failures

With the approach described above, we essentially lifted the formality of HRAs to the same level the remaining process of safety engineering has already. Considering multiple service failures as hazards and coping with the emerging complexity goes one step beyond this and advances the state of the art even more.

Multiple service failures can obviously occur in the case of common causes. But even if functions are specified to be independent, the integrity of their independence is given by the risk implied by both functions failing at the same time. Thus, one cannot argue omission of any hazards from the analysis, such as those consisting of multiple service failures. However, this implies that one needs to analyze an exponential number of hazards ($2^{\#\textit{single service failures}}$).

In order to cope with this challenge, we rely on computer-aided support for systematically reducing the number of hazards one needs to assess manually without omitting any hazard. ARID's fundamental idea is to reuse knowledge from already assessed risks (from single service failure assessments, for example) and, based on this, decide (automatically) whether we can exclude the un-assessed hazards from the manual assessment process.
In order to achieve this, we use component-integrated component fault trees as a formal representation of the system. Based on these models, we constructed algorithms that enable us to
(1) store risk assessment results and
(2) use the stored risk assessment information to reason about the necessity of further assessments.
The basic idea is to assume that each safety goal (derived from a risk assess-

ment) is (quantitatively) achieved, i.e., the failure rate of each minimal cut set is lower than or equal to the maximally allowed boundary (which is given by the safety goal). To make the decision about whether a hazard needs to be assessed or not, we perform fully automatic analyses with the previously determined failure rates (of the safety goals) as an information basis. The analysis result is a failure rate we can already assure with the given set of safety goals. If this failure rate is lower than the minimum we could produce with an assessment, there is no need to perform the assessment.

With the help of two case studies, we were able to show the feasibility of ARID, i.e., to ensure that all hazards are analyzed (either manually or fully automatically). We could further show that considering multiple service failures is indispensable: We showed that the safety goals derived from a single service failure analysis are insufficient and that critical parts of the system would not be handled adequately in the subsequent safety engineering process. With ARID, we have shown that considering multiple service failures as hazards is feasible and is indispensable from a safety engineering point of view.

5.1.3 Overall summary

The overall challenge we addressed in this thesis was to come up with a more structured approach for hazard analyses and risk assessments. The advancement of the state of the art we provided comprises two steps: The first step was to make the approach currently used in the automotive domain more formal in the sense that the process is less expert-dependent and that a higher degree of reuse is achieved. The second step was to advance the state of the art even further by widening the scope of hazards from single service failures to multiple service failures.

5.2 Reflection

In this section, we state lessons learned during our work and reflect on the topic of hazard analysis and risk assessments.

Our goals were to make the traditional HRA more consistent and efficient on the one hand, and, on the other hand, to raise the HRA to the next level by integrating multiple service failures into the HRA. For both goals, a more formal representation of HRAs was necessary. Our solution was to translate the overall process into a conceptual model-based representation as an interacting Four-Variable-Model.

From model-based HRAs to HRA@runtime?

We see this formalization as one fundamental step for handling hazard- and risk-related issues in the future. Novel ideas such as product line HRAs or runtime HRAs need a certain amount of formalism to be realized. For product line HRAs, for example, we think that the aspects of the system can be made variable and a decision modeler would need to be able to derive the respective interactions with the Human-FVM and the Environment-FVM that result from an actual product (system variant).

Another enabled future idea is to perform HRAs at runtime. If we consider emerging systems of systems and assume that these systems perform safety-critical functions collaboratively, we see that one cannot predict all possible emerging functions and hazards upfront! Thus, it seems reasonable to move at least parts of the HRA to run-time. We assume that an HRA at design time is difficult to do for these systems or would result (due to worst-case assumptions) in limited functionality. By putting the HRA on a model-based foundation, we therefore provided a first important steps towards "HRA@runtime".

However, one drawback of the current realization is the unformalized handling of severity and controllability. For our approach, we used the concepts of ISO 26262 to come up with severity values. This approach does not consider multiple injuries and there is nothing like an injury algebra to mathematically combine injuries into one severity level. A similar aspect is the concrete model of human capabilities. We used a well-known approach (Rasmussen's SRK model) and formalized this as a black box, but more as an extension showcase than as an actual model-based human capability framework. It would be important to further formalize the concrete behavioral capability model and store the knowledge of driving tests and field experience more formally.

However, one drawback of the current realization is the unformalized handling of severity and controllability. For our approach, we used the concepts of ISO 26262 to come up with severity values. This approach does not consider multiple injuries and there is nothing like an injury algebra, to mathematically combine injuries to one severity level. A similar aspect is the concrete model of human capabilities. We used a well-known approach (Rasmussen's SRK model) and formalized this as a black-box but more as an extension showcase than an actual model-based human capability frame-

work. It would be important to further formalize the concrete behavioral capability model and store the knowledge of driving test and field experience more formally.

Both aspects address the concrete formalization approach of GOBI and are handled using a practical approach based on ISO 26262.

Hazard-centric vs. unwanted-event-focused HRA

Another aspect we believe is worth putting research activities into is to think about whether it is beneficial to change the HRA from hazard-centric to accident- (unwanted event) focused. Especially if one transitions from functional safety to system safety, the aspects to focus on are primarily the unwanted events. These (including their occurrence likelihood) define whether or not a system is safe. The current ISO 26262 approach (based on functional safety) is hazard-centric and the criticality of a hazard is given by the occurrence likelihood and the severity of harm. With GOBI's formalization of the concepts we showed that a hazard is defined by a specification or deviations of a specification. The definition of safety, however, is not dependent on the specification, but on the concrete behavior. Thus, we claim that independent of the specification or the intention of an engineer, a system has a behavior (which includes unwanted behavior) and we, as safety engineers, need to make the complete system behavior safe. There is nothing like "it's not a hazard, it's a feature" as an adaptation of the quote "it's not a bug, it's a feature" in safety engineering.

Our idea of an unwanted event (accident) focused approach allows focusing on the system behavior. During the work on this thesis, the concept of unwanted event-centric HRAs re-emerged in different industrial projects (which were independent from this PhD). For example, one addressed autonomous driving (or highly innovative driving assistance systems). One inherent question of these systems is: Which behavior is safe, and which is not safe? Thus, a behavior-centric approach would not work. The only change was to use an unwanted-event-focused solution approach.

The potential of our operational situation formalization

For an HRA@runtime, one would require a complete model of operational situations and their exposure assessment. For this it would be necessary to model the complete domain knowledge with respect to operational situations. With OASIS and HEAT, we introduced an ontology-based, knowledge-conserving approach for influence factors, operational situations, and domain knowledge. We showed the suitability of this approach. In order to obtain an (almost) complete model, automotive companies should collaborate on one integrated domain knowledge model.

As already stated in the introduction, especially the exposure of operational situations is not a USP for any company. Companies already try to synchronize their work (manually). With OASIS and HEAT, we provided a common platform for operational situations (and their exposure assessments) and thereby make individual HRAs more consistent and the process more efficient. An institutionalized, central knowledge base using OASIS and HEAT would therefore be easily possible.

The asymptotic completeness of (relevant) operational situations

When we thought about completeness of operational situations, we tried to get an idea of how many situations we are talking about when we talk about completeness. As described in the respective section, our current ontology includes, over nine industrial HRAs and has shown its usefulness. The more HRAs the ontology contains, the higher the gain wrt. the reuse of situations and completeness. We argued as part of our solution approach that a complete list of operational situations cannot be built. With the lessons learned from the nine industrial case studies, we believe, however, that a complete list of practically *relevant* operational situations can be built. If we assume a simple asymptotic increase of knowledge and further assume that we have currently covered about 50% of all situations, and that only 99% of all situations are practically relevant, we would need around 60 real-world industrial HRAs to build our complete knowledge base! The following mathematical calculation reflects this:

$$\begin{aligned} \rightarrow 50\% &= 1 - e^{-\frac{9}{x}} \;\; \Rightarrow x \approx 13 \\ 99\% &= 1 - e^{-\frac{y}{13}} \;\; \Rightarrow y \approx 60 \end{aligned}$$

This equation is based on many assumptions (such as the exponential, asymptotic growth); however, our goal was to get an idea in terms of order of magnitude. For this we believe the estimation of roughly 60 HRAs is realistic. This, in turn, would mean that a practically relevant (complete) fraction of influence factors consists roughly of 400 influence factors. For an ontology and for reasoning, this is a very small "problem" space, i.e., feasible for a practical application.

Transition from single service failures to multiple service failures

Another more serious problem for future HRAs (and especially a HRA@runtime) is the challenge of transitioning from single service failures to multiple service failures as relevant hazards. With ARID we have developed the first approach to dealing with those hazards. In the case studies we performed during our work, ARID worked very well. But we proved that the problem is NP-equivalent. Thus, it would be interesting to have more and more complex industrial case studies to really prove ARID's practical applicability.

The main fact that became evident for us during our work on ARID is that ARID's problem complexity strongly depends on the complexity of the system. We claimed that if ARID cannot deliver a guarantee for completeness of the hazard analysis, the engineer has lost control over the complexity of the system. Even nowadays it is common that system architectures are changed if the safety engineer cannot ensure the required risk reduction. We think that safety engineering should have this "veto" already during the very first step, the HRA! If one cannot prove completeness of the hazard analysis (due to the complexity of the functional interrelations), one should change/break these interrelations such that the system's hazards can be completely analyzed. Our general claim is therefore: Don't build a system for which you cannot ensure 100% hazard analysis coverage!

5.3 Future Work

In this section, we will present four possible future work areas based on this thesis.
The first is to create a product line HRA approach.
The second is an extension to the first. The idea would be to integrate ARID into the decision modeler to produce only 100% analyzable systems.
The third is to move from functional safety to system safety and adapt the HRA accordingly.
The fourth is to transfer this to another domain.

Currently, GOBI and the subsequent refinements by OASIS and HEAT are a knowledge base that can be used for concrete HRAs. However, if there were multiple HRAs or different versions of the system for which to perform an HRA, one would need to apply the HRA process (with SAHARA support) to each single HRA. Product line approaches deal with this challenge on the system level by integrating all variants in a 200% model and use a decision/variation modeler to "extract" concrete versions from the 200% model. As future work, it would therefore be interesting to investigate a 200% HRA model and use this model in conjunction with the 200% system model, to not only reuse and derive versions of the system, but additionally their respective HRAs. To do so, one would need to identify variable and constant elements in the HRA model. One would furthermore need a kind of HRA decision dependency model to ensure consistent and correct extraction of HRAs.

The second idea is a continuation of the product line HRA idea. As discussed in the reflection, one should use ARID in the investigation of the design space. A 200% model and the decision model rules for extracting concrete systems represent a structured representation of the design space. It would therefore be a great advantage if ARID could be applied to or integrated into the 200% model. By doing this, we could restrict the design space to systems for which we can ensure 100% coverage of hazard analyses. One important prerequisite would be a product line fault tree (or failure propagation) approach. However, at the moment we cannot say whether the existing approach is sufficient to be used in a product line approach for ARID. If we assume that this is given, the next challenge would be to apply the iterative knowledge capturing (as quantitative prime implicants knowledge) in a product line environment. This would require deciding for each piece of knowledge to which version this information contributes and to which it does not. Thus, one would need to define for each piece of knowledge an additional rule in the decision modeler (or similar variability information).
Another inherent problem of a product line approach could be problem complexity. We claimed that one should only build a system for which 100% hazard analyses can be ensured. If we build a 200% model, the complexity is obviously much higher, and we want to restrict the concrete systems to be analyzable. This, however, introduces complexity into the step of deriv-

ing the 200% ARID model. One would probably need an iterative decision process in the sense that ARID is not fully applied to the 200% model, but only some parts. Then a first level of decision resolution is made and the resulting 150% model is the basis for the next steps of ARID and so forth. However, this is only a rough solution idea of the author and one would need to investigate the problem in more detail and derive a concrete solution approach based on a good understanding of the problem space of product line ARID applications.

The fourth area of future work addresses our current focus on automotive systems. The idea is to transfer this formalization approach to other domains. If we consider domains that are similar to the one of automotive, such as commercial vehicles, we assume that the transfer would be pretty straightforward. However, if we try to migrate this idea to domains such as avionics, or medical devices, we see some challenges arising. One challenge is that in medical devices, there does not exist a risk graph approach; thus, no parameters (like E, S, C) are defined. One would have to come up with these essential parameters. Furthermore, formalizing medical operational situations, like we did for automotive is much more difficult for medical devices because the variance in medical devices is much higher than in automotive systems. One medical device is a clinical thermometer, another is a brachytherapy device, and a third is an X-ray device. The "environment" and the environmental assumptions are very different compared to the automotive domain. One idea would be to classify the medical devices first and derive an operational situation domain model for each class. Whether this is feasible is an open question, however.

References

[ADB$^+$99] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, 1999. Springer-Verlag.

[ADH$^+$10] Rasmus Adler, Dominik Domis, Kai Höfig, Sören Kemmann, Thomas Kuhn, Jean-Pascal Schwinn, and Mario Trapp. Integration of Component Fault Trees into the UML. In Jürgen Dingel and Arnor Solberg, editors, *MoDELS Workshops*, volume 6627 of *Lecture Notes in Computer Science*, pages 312–327. Springer, 2010.

[All08] Christian Allmann. *Situations- und szenariobasiertes Anforderungsmanagement in der automotive Elektronikentwicklung*. Cuvillier, 2008.

[AP94] Agnar Aamodt and Enric Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Commun.*, 7(1):39–59, 1994.

[AS97] Varol Akman and Mehmet Surav. The Use Of Situation Theory In Context Modeling. *Computational Intelligence*, 13:427–438, 1997.

[ATE13] ATESST2 Project. `http://www.atesst.org`, August 2013.

[Ban75] Richard Bandler. *A book about language and therapy.* Science and Behavior Books, 1975.

[Bar08] Jonathan Baron. *Thinking and deciding*. Cambridge University Press, 4 edition, October 2008.

[Ber03] Kurt-Jürgen Berger. *Technologie Kraftfahrzeugtechnik Grund- und Fachbildung*. Bildungsverl. EINS - Gehlen, 2003.

[Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing 2011 edition, October 2007.

[BOHL74] S. P. Baker, B. O'Neill, W. Haddon, and W. B. Long. The injury severity score: a method for describing patients with multiple injuries and evaluating emergency care. *The Journal of trauma*, 14(3):187–196, March 1974.

[Bro03] Frank M. Brown. *Boolean reasoning : the logic of Boolean equations*. Dover Publications, 2003.

[Cry08] David Crystal. *A dictionary of linguistics and phonetics*. Blackwell Pub., 2008.

[Dav03] John Davies. Safety management a qualitative systems approach, 2003.

[Dev95] Keith J. Devlin. *Logic and information*. Cambridge University Press, 1995.

[Dev06] Keith Devlin. *Situation theory and situation semantics*, volume 7 of *Handbook of the History of Logic*, pages 601–664. Elsevier, 2006.

[DH89] D. Drusinsky and D. Harel. Using statecharts for hardware description and synthesis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 8(7):798–807, 1989.

[Dom12] Dominik Domis. *Integrating fault tree analysis and component-oriented model-based design of embedded systems*. PhD thesis, 2012.

[fAR11] Center for Automotive Research. Automotive Technology:Greener Vehicles, Changing Skills:ELECTRONICS, SOFTWARE & CONTROLS REPORT. Technical report, Center for Automotive Research, 3005 Boardwalk, Suite 200, Ann Arbor, MI 48108, May 2011.

[GIMT08] Andy Galloway, Frantz Iwu, John McDermid, and Ian Toyn. On the Formal Development of Safety-Critical Software. In Bertrand Meyer and Jim Woodcock, editors, *Verified Software: Theories, Tools, Experiments*, volume 4171 of *Lecture Notes in Computer Science*, pages 362–373. Springer Berlin Heidelberg, 2008.

[Gra93] Jeffrey O. Grady. *System requirements analysis*. McGraw-Hill, 1993.

[Gri76] John Grinder. *The structure of magic. Vol 2, Vol 2,*. Science and Behavior Books, 1976.

[Har87] David Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.

[Hen80] K. L. Heninger. Specifying Software Requirements for Complex Systems: New Techniques and Their Application. *IEEE Trans. Softw. Eng.*, 6:2–13, January 1980.

[HJE$^+$10] C. Haasper, M. Junge, A. Ernstberger, H. Brehme, L. Hannawald, C. Langer, J. Nehmzow, D. Otte, U. Sander, C. Krettek, and H. Zwipp. Die Abbreviated Injury Scale (AIS). *Der Unfallchirurg*, 113(5):366–372, May 2010.

[Hol93] Erik Hollnagel. The phenotype of erroneous actions. *Int. J. Man-Mach. Stud.*, 39(1):1–32, July 1993.

[HRWS06] J. Hartmann, S. Rittmann, D. Wild, and P. Scholz. Formal incremental requirements specification of service-oriented automotive software systems. In *Service-Oriented System Engineering, 2006. SOSE '06. Second IEEE International Workshop*, pages 130–133. IEEE, October 2006.

[HWL06] Erik Hollnagel, David D. Woods, and Nancy Leveson. *Resilience engineering : concepts and precepts*. Ashgate, April 2006.

[Ira11] Pierre Iraguha. Evaluation of the ARID Process. Master's thesis, 67653 Kaiserslautern, October 2011.

[ISO11] ISO 26262 Road vehicles - Functional safety. Technical report, International Organisation for Standardisation, 2011.

[KB13] Kraftfahrt-Bundesamt. Jahresbilanz des Fahrzeugbestandes am 1.Januar 2013. `http://www.kba.de/nn_125264/DE/Statistik/Fahrzeuge/Bestand/bestand__node.html?__nnn=true`, August 2013.

[Kle92] Trevor A. Kletz. *Hazop and hazan : identifying and assessing process industry hazards*. Institute of Chemical Engineers ; Distributed exclusively in the USA and Canada by Hemisphere Pub. Corp., 1992.

[KT72] Daniel Kahneman and Amos Tversky. Subjective probability: A judgment of representativeness. *Cognitive Psychology*, 3(3):430–454, July 1972.

[KT11] Sören Kemmann and Mario Trapp. SAHARA - A Systematic Approach for Hazard and Risk Assessment. Technical Report 2011-01-1003, SAE, 2011.

[Lea94] David B. Leake. Case-based reasoning. *The Knowledge Engineering Review*, 9(01):61–64, 1994.

[Lev00] Nancy G. Leveson. *Safeware : system safety and computers : [a guide to preventing accidents and losses caused by technology]*. Addison-Wesley, 2000.

[Lew43] Kurt Lewin. *Defining the 'field at a given time'*. 1943.

[Lig09] Peter Liggesmeyer. *Software-Qualität : Testen, Analysieren und Verifizieren von Software*. Spektrum, Akad. Verl., 2009.

[Lut01] Robyn R. Lutz. Analyzing Software Requirements Errors in Safety-Critical, Embedded Systems, 2001.

[Mil56] George A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[MT01] Steven P. Miller and Alan C. Tribble. Extending the four-variable model to bridge the system-software gap. In *Digital Avionics Systems, 2001. DASC. 20th Conference*, volume 1, pages 4E5–1. IEEE, 2001.

[MzH04] Michael Meyer zu Hörste. *Methodische Analyse und generische Modellierung von Eisenbahnleit- und -sicherungssystemen*. PhD thesis, 2004.

[PD01] Rubén Prieto-Diaz. Reuse in Engineering vs. Reuse in Software: Why Are They Incompatible? In *Symposium on Software Reusability*, 2001.

[PM95] David L. Parnas and Jan Madey. Functional documents for computer systems. *Sci. Comput. Program.*, 25:41–61, October 1995.

[PM99] Yiannis Papadopoulos and John McDermid. Hierarchically Performed Hazard Origin and Propagation Studies. In Karama Kanoun, editor, *Computer Safety, Reliability and Security*, volume 1698 of *Lecture Notes in Computer Science*, page 688. Springer Berlin / Heidelberg, 1999.

[Ras82] Jens Rasmussen. Human errors. A taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents*, 4(2–4):311–333, 1982.

[Rea90] J. T. Reason. *Human error*. Cambridge University Press, 1 edition, October 1990.

[RgDaGc$^+$06] Juan A. Recio-garcía, Belén Díaz-agudo, Pedro González-calero, Antonio S. Ruiz Granados, and Dep S. Informáticos. Ontology based CBR with jCOLIBRI. In *Applications and Innovations in Intelligent Systems XIV*, pages 149–162. Springer-Verlag London, 2006.

[Rup04] Chris Rupp. *Requirements-Engineering und -Management : professionelle, iterative Anforderungsanalyse für die Praxis*. Hanser, 3., neu bearb. a. edition, 2004.

[Sas96] Tsutomu Sasao. *Representations of discrete functions*. Kluwer Academic, 1996.

[SLP04] Thomas Strang and Claudia Linnhoff-Popien. A Context Modeling Survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

[Stä11] T. Ständer. *Eine modellbasierte Methode zur Objektivierung der Risikoanalyse nach ISO 26262*. 2011.

[vSPM93] A. J. van Schouwen, D. L. Parnas, and J. Madey. Documentation of requirements for computer systems. In *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pages 198–207. IEEE, January 1993.

[WBS10] Petra Winzer, Friedrich-Wilhelm Bach, and Eckehard Schnieder. Sicherheitsforschung-Chancen und Perspektiven, 2010.

[wik13] Hazard (risk) in Wikipedia. `http://en.wikipedia.org/wiki/Hazard_(risk)`, August 2013.

[WMP$^+$10] Martin Walker, Nidhal Mahmud, Yiannis Papadopoulos, Fulvio Tagliabo, Sandra Torchiaro, Walter Schierano, and Henrik Lönn. Review of relevant Safety Analysis Techniques. Technical report, The ATESST2 Consortium, 2010.

[ZJ97] Pamela Zave and Michael Jackson. Four dark corners of requirements engineering. *ACM Trans. Softw. Eng. Methodol.*, 6(1):1–30, January 1997.

Lebenslauf

Name	Sören Kemmann	
Schulbildung	1987-1991	Dekan-Ernst Grundschule Grünstadt
	1991-2001	Leininger-Gymnasium Grünstadt Abschluss: Abitur (Note: 1.0)
Studium	2001-2007	Technische Universität Kaiserslautern Abschluss: Diplom-Informatiker (Note: 1.0)
Berufstätigkeit	2007-2011	Wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Experimentelles Software Engineering, Kaiserslautern
	2011-2015	Abteilungsleiter "Embedded Systems Quality Assurance" Fraunhofer-Institut für Experimentelles Software Engineering, Kaiserslautern
	2015-heute	Geschäftsführer B+B Unternehmensberatung GmbH & Co. KG Bad Dürkheim

Kaiserslautern, den 03.03.2015