CHALMERS
UNIVERSITY OF TECHNOLOGY

Requirement representation for safety-critical and fairness aware automotive perception systems

Identifying requirements representation challenges for multi-party collaboration.

Master's thesis in Software engineering and technology

OSKAR JAKOBSSON
ZUZANA ROHACOVA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Requirement representation for safety-critical and fairness aware automotive perception systems

Identifying requirements representation challenges for multi-party collaboration

OSKAR JAKOBSSON
ZUZANA ROHACOVA

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2024

Requirement representation for safety-critical and fairness aware automotive perception systems
Identifying requirements representation challenges for multi-party collaboration
OSKAR JAKOBSSON ZUZANA ROHACOVA

Supervisor: Hans-Martin Heyn, and Hina Saeeda Department of Computer Science and Engineering
Examiner: Birgit Penzenstadler, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Requirement representation for safety-critical and fairness aware automotive perception systems
Identifying representation challenges for cross company collaboration
OSKAR JAKOBSSON & ZUZANA ROHACOVA
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Background: Advancements in the field of machine learning (ML), have unlocked new capabilities for Driving Automation Systems (DAS). These DAS systems rely on the input from automotive perception systems. These systems exist in a safety-critical domain, and well-defined requirements are key to ensure that they can operate safely. However, requirements engineering (RE) for ML-enabled systems has been identified as a challenge in research. **Aim:** The thesis aimed to investigate current approaches in RE for automotive perception systems, and identify what challenges exist and which processes work well. Specifically, approaches for requirement representations, model kinds, templates, and structures. The thesis also wanted to explore if a shared language, in regards to domain description, reference system architecture, and reference information model, could help mitigate potential challenges without hindering approaches that work well. **Methods:** An exploratory case study was conducted by interviewing experts in the automotive field. This included participants from an automotive OEM, suppliers to said OEM and experienced researchers in the automotive field. In total ten interviewees were consulted. **Results:** The challenges and what works well in the current processes in the case study companies were identified through thematic analysis of the interview data. The thesis explored the potential of a shared language to mitigate these challenges by discussing the topic with interviewees and observing brainstorming workshops for the creation of the shared language. **Conclusions:** The thesis shows that there is a lack of industry standards in RE for ML-enabled automotive perception systems, which complicates multi-party development. According to interviewees, the shared language has potential to alleviate the identified challenges. However, the feasibility of the shared language is still unclear.

Keywords: Requirements engineering, requirements, perception systems, thesis, requirements representation, safety-critical, multi-party development, Automotive, shared language.

Acknowledgements

We would like to thank our supervisors, Hans-Martin Heyn and Hina Saaeda for their continued support and valuable feedback throughout the thesis. Their vast knowledge and expertise has been incredibly helpful. We would like to thank all members of the FAMER project who helped guide us in our study. We would also like to extend our deepest appreciation to the participants of the interview study, who took time out of their day to share their experience and expertise. A final thanks to our examiner Birgit Penzenstadler for evaluating our thesis and giving valuable feedback.

Oskar Jakobsson & Zuzana Rohacova, Gothenburg, May 2024

A personal thanks to my partner Ida, for unending emotional support throughout my academic journey. I am blessed to have you in my life.

Oskar Jakobsson, Gothenburg, May 2024

I want to thank my friends and family for their support in my academic journey.

Zuzana Rohacova, Gothenburg, May 2024

Contents

List of Figures

List of Tables

1
Introduction

When developing automotive perception systems, it is common to collaborate not only across multiple disciplines of engineering, but also across multiple companies [1]. The owner and developer of the system may have multiple suppliers delivering technological solutions and components with varying degrees of complexity. Furthermore, automotive perception systems are often used by driving automation systems (DAS). DAS is a safety-critical domain, and as such the delivered components and potential accompanying software must adhere to strict rules and regulations [2]. During such conditions, it is critical to have well defined requirements [3]. A well defined requirement is unambiguous, testable or measurable, traceable, feasible and understandable [4, 5]. Well defined requirements do not only ensure that the delivered product is of high quality and adheres to relevant safety standards, it also helps to avoid potential disagreements and arguments regarding the delivered product [6, 7].

Furthermore, it is becoming increasingly more common for automotive perception systems to be used in combination with ML-based components or other types of data intensive systems [8]. As such, requirements for data annotation and data quality naturally arise, which research has found to be particularly difficult to define [9, 10].

Another common challenge when dealing with multi-party collaboration is the differing internal structures and communication standards [11]. As such, domain-specific tacit knowledge may appear obvious to the customer and product owner. However, the supplier may not have the domain experience to ascertain the exact needs of the customer without them being explicitly stated.

Studies have shown that requirements engineering (RE) for automotive perception systems is a major challenge [12, 13]. However, challenges specific to requirements representation in such systems have not been explored in depth. The aim of the thesis is therefore to contribute to this body of work by identifying challenges in requirements representation, model kinds, templates, and structures for automotive perception systems. The thesis also aims to identify what current approaches related to these aspects of RE currently work well. As these systems are commonly developed in a multi-party context, collaboration aspects across organisations will also be considered. Moreover, the thesis aims to evaluate whether a shared language could help to mitigate potential challenges, without interfering with processes that currently work well.

1.1 Statement of the problem

The advances in ML has presented the automotive industry with new potential for DAS, which encompasses both autonomous driving (AD) and advanced driver assistance systems (ADAS) [8]. However, the advances in ML have also brought with it a new set of challenges [14, 15]. These systems, crucial for enhancing road safety and driver convenience, demand meticulous attention to their design, development, and validation processes [16]. Furthermore, the complexity of safety-critical systems coupled with the incorporation of ML technologies poses challenges in RE [17, 18].

The challenge of RE is amplified in the context of multi-party collaborations involved in the development of automotive perception systems [1]. These collaborations bring together diverse expertise, perspectives, and methodologies, thereby making it challenging to establishing cohesive requirements frameworks [19]. As a result, achieving consensus on critical requirements becomes a difficult task, hindered by differences in terminologies, priorities, and understandings across organizational boundaries [20].

The difficulty in defining requirements for these automotive perception systems is due to three key challenges: the inherent complexity of safety-critical systems, the integration of ML technologies, and the collaborative nature of development involving multiple companies [1, 21, 22]. Furthermore, due to ML being integrated into these systems, creating well defined fairness requirements are becoming increasingly relevant [23, 24]. An unfair ML-system may contain biases, which may discriminate against groups of people or objects in their calculations [25].

1.2 Purpose of the study

The purpose of this thesis is to investigate current approaches and processes for requirements representation for automotive perception systems. Through a case study, practices, methodologies, and stakeholder perspectives of an automotive supply chain are examined. The aim is to identify challenges and what works well in the current approaches and processes. Furthermore, the thesis aims to evaluate whether a shared language can alleviate potential challenges, without hindering the approaches that currently work well.

1.3 Research questions

To investigate the aforementioned concepts, the thesis will aim to answer the following research questions:

RQ1: What requirements representations, model kinds, templates, and structures work well for defining requirements related to safety and fairness for the case companies?

The aim of this research question is to get a deeper understanding of what currently works well for the case study companies. This will be accomplished by conducting interviews and performing thematic analysis on the gathered data.

RQ2: What are the challenges in finding common requirements representations, recommending model kinds, templates, and structures for defining requirements related to safety and fairness across the case companies?

The aim of this research question is to identify what approaches challenges currently exist within the current approaches for the case companies. This will be accomplished by conducting interviews and performing thematic analysis on the gathered data.

RQ3: How can a shared language in regards to domain description, reference system architecture, and reference information model help to overcome previously identified challenges?

A field study will be conducted to evaluate if the shared language could help alleviate potential challenges, without conflicting with or hindering processes that already work well.

1.4 Facilitating Multi-Party Engineering of Requirements

This thesis is a part of a larger research project called Facilitating Multi-Party Engineering of Requirements (FAMER) [26]. FAMER is a research project spanning three years and containing actors from both industry and academia. The goal is to establish *"...concepts, models, and techniques of effectively building requirements knowledge for safe perception systems."*. The project is divided into five distinct work packages, where each package pertains to a certain part of the requirements engineering process. This thesis is placed in work package two (WP2), which pertains to requirements representation. The shared language that the thesis will be evaluating is from the previous work package (WP1).

1.5 Scope and limitations

The thesis will focus solely on one particular automotive OEM and two of its suppliers, which together form a supply chain for automotive perception systems. As such, it should be classified as a case study, and therefore it will not guarantee generalizability. Furthermore, the thesis will not provide a holistic solution to previously mentioned requirements challenges. Although workflow of the OEM and its suppliers will be considered, this workflow might not necessarily map directly to requirement engineering phases such as elicitation, verification and validation. The thesis will focus solely on requirement representation and evaluating a shared language to communicate safety and fairness requirements for automotive perception systems. Safety is a very prevalent quality requirement when it comes to the development of software for vehicles. Furthermore, the necessity of ensuring a fair and unbiased system becomes even more relevant as more data-intensive and ML-based components are being integrated into these automotive perception systems [27, 24]. As such, whether the shared language is suitable for any other context is not of importance and will not be considered.

2

Background

This section presents some of the necessary background to grasp the topics of this thesis. Firstly, safety-critical automotive systems are explained, including two examples of such systems. Secondly, a brief description of what an operational design domain (ODD) is given. Finally, the concept of non-functional requirements is explained, with more detail given for safety and fairness requirements.

2.1 Safety-critical automotive systems

A safety-critical system is a system where a failure or malfunction could result in catastrophic consequences, including loss of life, severe injuries, loss or severe damage to equipment/property or environmental damage [28, pp.339-342]. Because the consequence of a failure could be detrimental, the development of safety-critical systems typically follows stringent engineering practices and adhere to industry standards [3]. One domain that has historically dealt with development of safety-critical systems is the automotive industry. As such, RE, especially when it comes to defining safety requirements, has been thoroughly researched in the context of automotive systems [3, 16, 12]. Furthermore, there are ISO standards pertaining to the development and requirements of safety-critical systems. For example *ISO 26262*, which defines standards for the functional safety of road vehicles [29].

DAS & Automotive perception systems

Recent advances in ML have led to both increased popularity and technical feasibility of DAS systems. An example of an ADAS function is automatic emergency braking (AEB) [9]. To implement AEB, the system must be able to perceive objects, understand the distance and dimensions to and of said object, and based on various parameters decide whether emergency braking is required. These DAS systems are often developed and deployed with the help of various ML-based components [30]. Furthermore, DAS require input that represents the outside world to be able to perceive and perform calculations that result in actions being taken by the system [12]. These input-systems are called automotive perception systems.

An automotive perception system is a system that can perceive environmental surroundings through the use of sensors such as vision cameras, light detection and ranging (LIDAR) and radio detection and ranging (RADAR) [31]. Moreover, ML-based perception systems fuse the sensor inputs using multi-modal sensor fusion to analyze the vehicle environment [12, 32]. It should be noted that there exists non-ML based perception systems [33, pp.3-8]. However, in this thesis the focus will be on ML-based automotive perception systems.

Operational design domain

Operation design domain (ODD) is used to define under which specific conditions a DAS system is expected to be able to operate safely [34]. Characteristics such as speed, type of road and weather conditions are used to define the capabilities and limitations of the system [34].

2.2 Non-functional requirements

The engineering of safety-critical systems necessitates careful consideration of requirements because these systems rely on various components and computing systems. Failure to adhere to RE methods and practices may lead to poorly defined requirements, incorrect specifications, and flawed structures, rendering complex systems more susceptible to failures [35, 16].
These systems should fulfill different requirements - functional and non-functional. While functional requirements describe what the system should do, non-functional requirements describe the properties, qualities, and constraints of the system [36].

Safety requirements

Safety can be defined as the absence of failures or conditions that render a system dangerous [22]. Systems undergo certification according to relevant safety standards to guarantee safety. For instance, in the automotive industry, the *ISO 26262* [29] standard is adhered to for achieving functional safety. Requirements play a crucial role in specifying constraints to support system safety. By defining proper safety requirements and adhering to safety standards, the associated risks with the system are significantly reduced [37].

For safety-critical systems, safety requirements are a top priority and should be considered already at the start of a development project. Considering safety later in the project is expensive as it can alter the architecture of the system [16, 38]. However, modern development practices, such as agile project management, are becoming quite popular even in companies that focus on the development of safety-critical systems [35, 12]. Especially with the introduction of ML components, precise safety requirements may not be clear until the end, and they can be modified during the development.

Fairness requirements

Fairness can be defined as the ability of a system to operate in a fair and unbiased manner [22].Data can contain biases that can influence system's algorithms and make unfair and biased decisions [39]. These decisions of the unfair system can then lead to discriminatory behavior of the system. Exploring fairness became relevant in ML-enabled systems as they take into large quantities of data. Looking into fairness properly helps to ensure a fair and trustworthy ML-enabled system [23].

Additionally, with the development of ML and especially with the advancement of Artificial Intelligence (AI), different standards are being developed. For example, the EU AI Act [40]. In the Act, the focus is on the fair principles and processes related to AI and ML-enabled systems.

3

Related work

This chapter presents the current state of related research in the field of requirements engineering for automotive perception systems. Firstly, traditional processes and approaches for requirements engineering are presented. Thereafter, the previously studied issues of RE for ML-enabled automotive systems are explored and explained. Finally, the challenges related to RE for multi-party collaboration are presented.

3.1 Traditional processes of RE

RE is a critical phase in software development that defines the needs of stakeholders for a given system, and specifies what the system must be able to do to fulfill that need [4]. This is done by eliciting, analyzing, documenting, and managing software requirements. Over the years, various processes and methods have been established to guide practitioners through this task [4, 41, 42]. In this section, some examples of these processes and their methods of requirements representation will be presented. Furthermore, the section will expand upon how these processes relate to multi-party development scenarios, where multiple companies collaborate to develop and maintain a unified product.

The different phases of RE

Requirements engineering is typically divided into the following phases [42]: **Inception, Elicitation, Elaboration, Negotiation, Specification, Verification & Validation** and **Management & Maintenance**

In this section each phase will be presented in sequential order. It is worth noting however, that these phases do not necessarily have to be executed and finished sequentially, nor is it common for them to do so in practice [42]. More commonly, engineers and developers iterate over these phases as new information is discovered. As stated in [42, pp.68-69] "*In reality, the processes tend to be collaborative, cooperative, iterative, and incremental. By one hand, the cooperation and collaboration amongst the requirements engineers and between them and the stakeholders presupposes parallelism/concurrency in the execution of the tasks. By another hand, the iterative and incremental nature of the process means that a task previously executed can be executed again, as many time as needed.*". However, there is of course some dependencies between the phases, as it would be impossible to validate a requirement in the validation phase without having a defined requirement to validate.

The **inception** phase is the initial inspiration of a project or ideation of a product [43]. Usually this happens through the identification of some business need. It is also common for the inception to arise due to some level of frustration or annoyance in relation to a current practice or product [42]. During this phase stakeholders define a business case and an estimated scope of the product/project [42]. Furthermore, [43] defines the inception phase as: "*... you establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.*".

Elicitation refers to the initial phase of requirements engineering, where the objective is to extract comprehensive and accurate information regarding the desired system from the relevant stakeholders [44]. This process involves engaging with stakeholders through various means such as interviews, surveys, and workshops to understand their needs, objectives, constraints, and expectations. Elicitation also includes uncovering tacit requirements that stakeholders may not articulate directly [44]. The success of this phase relies heavily on fostering an open and collaborative environment where stakeholders feel encouraged to express their perspectives and requirements candidly [42, 44]. The elicitation phase serves as a critical starting point for the subsequent analysis and specification stages [41].

The **elaboration** phase pertains to examination and refinement of the gathered requirements to ensure clarity, coherence, and completeness [43]. This phase entails categorizing requirements into distinct types and scrutinizing them for inconsistencies, ambiguities, or conflicts [42].

During the **negotiation** phase the impact of the requirements on project parameters such as cost, schedule, resources, and technical feasibility is assessed to ensure alignment with project objectives [43, 42]. By also evaluating the technical feasibility of requirements, the negotiation phase enables informed decision-making and helps mitigate the risk of project failure [43].

The **specification** phase constitutes the formalization of elicited requirements [43]. This phase involves translating stakeholder needs and expectations into unambiguous and detailed requirements that can be understood and implemented by development teams. Requirements are documented and represented using one of or a combination of methods. Such as natural language, diagrams, and formal notations, facilitating clear communication and interpretation [42]. The resulting requirements specification document serves as an agreement between stakeholders and development teams, guiding the subsequent phases of system design, implementation, and testing.

The **verification & validation** phase aims to deal with two things. On the one hand, the phase aims to *verify* that the specified requirements accurately represent the needs and expectations of stakeholders [43, 42]. On the other hand, the phase also aims to *validate* that the requirements are specified in a correct manner [44]. The literature is somewhat divided in this. Some literature only talks about verification [42, 4] and some argue for both verification and validation [44, 43]. Nonetheless, the main goal of this phase is to ensure the requirements are represented in a correct way, and accurately capture the needs of the system [44, 43, 42, 4].

The **management and maintenance** phase involves ongoing tracking, documentation, and evolution of requirements throughout the project lifecycle [43]. This phase encompasses activities such as requirements traceability, version control, and change management to ensure that requirements remain aligned with evolving stakeholder needs and project objectives [42].

As previously stated these phases are not necessarily executed sequentially. Moreover, in certain process frameworks, some phases may receive more attention, whereas some receive less [45]. The next section will aim to elaborate on what process models are commonly used and how they relate to RE.

Common process models and their relationship with RE

The Waterfall Model is one of the oldest process models in software engineering [46, 47, 45]. It consists of distinct phases, meant to be completed sequentially, with *"some overlap and splash back acceptable between phases"* [47]. The phases are as follows: requirements analysis, design, development, testing, implementation, and maintenance [45]. The waterfall model allows for a clearly defined process where each phase has a distinct purpose, often with an accommodating deadline. This helps in making high-level planning and resource distribution simpler [45]. The waterfall model relies heavily on clear documentation and specification, further adding to the planning process [48]. Furthermore, as the phases are meant to be executed in sequential order, it is crucial that the specified requirements are correct and thoroughly thought out, since going to back to redefine the requirements will be difficult [46]. Additionally, no working software will be produced until late in the development life-cycle, making it challenging to spot potential mistakes early on [46]. Moreover, In multi-party development, the Waterfall Model may present challenges due to its linear nature, making it difficult to accommodate diverse inputs and requirements from multiple stakeholders and development teams [43].

The V-Model is an extension of the Waterfall Model that emphasizes verification and validation activities at each stage of development [47]. It correlates each development phase with a corresponding testing phase [45]. Research has found that the pros and cons of the V-model are similar to that of the waterfall model, with certain modifications [45]. For example, the V-model is a fairly straightforward and easy to use process model [46, 45]. However, it is also considered less flexible as compared to the waterfall model [47, 46, 45]. That being said, it is possible to alter the requirements during any point in the development life cycle, without having to restart the entire development process [45]. However, if requirements are changed midway, it is not only the requirements documentation that must be updated, but also the test documentation [45]. In multi-party development, the V-Model helps to ensure that the requirements of each party are adequately addressed and tested early, which may reduce the risk of misunderstandings and conflicts [46]. However, as the V-model is essentially an extension of the waterfall model, similar issues may arise during multi-party development [43].

Contrary to the sequential nature of the Waterfall and V-Model, Agile methodologies, such as Scrum and Extreme Programming (XP), focus on iterative and incremental development [48]. Requirements in Agile are often represented using user stories, which are concise descriptions of a feature from an end-user perspective [43]. This is commonly done in the style of the role-feature-reason template: *as a* ***[type of user]*** *i want* ***[some feature]*** *because of* ***[some reason]*** [49]. Agile practices like daily stand-up meetings and sprint reviews facilitate continuous communication and alignment among collaborating parties, ensuring that evolving requirements are effectively managed and implemented [43]. However, in larger projects it may become difficult to accurately estimate effort and time required for the project [45]. Agile methodologies also often lack extensive documentation, which in some cases can be seen as an issue [46].

3.2 RE for ML-enabled automotive systems

Communicating requirements for safety-critical systems has historically been considered a challenge [18, 50, 51]. This is due to multiple factors such as the complexity of safety standards, the interdisciplinary nature of safety-critical systems, and the need for precise language to convey critical information without ambiguity. Safety standards such as *ISO 26262* for automotive systems require meticulous attention to detail and adherence to rigorous processes to ensure the safety and reliability of the final product [29, 18]. The dynamic nature of technology and evolving safety regulations adds another layer of complexity, necessitating constant updates and revisions to safety requirements documentation. Despite these challenges, there have been advancements in communication methodologies. For example, model-based systems engineering (MBSE), formal methods and the use of a more precise natural language in requirement specification. These advancements offer promising solutions to enhance the clarity and effectiveness of safety requirements communication in safety-critical systems development [52, 53].

However, RE for ML is still considered a challenge in research [14, 54, 17]. One of the primary difficulties arises from the complexity and non-deterministic nature of ML algorithms, which makes it challenging to precisely define and articulate system requirements [55, 14, 56]. As ML algorithms learn and adapt based on data inputs, the behavior of these systems can be influenced by numerous factors, including data quality, distribution, and model architecture.

Furthermore, the latest version of the *ISO 26262* standard does not cover ML [57]. Currently, there are no universal standards for engineers to refer to when developing ML-enabled automotive perception systems. However, there is an ISO-standard that is currently under development, namely, *ISO/CD PAS 8800*, *Road Vehicles, Safety and artificial intelligence* [58]. However, the standard is not yet published and at the time of writing it is not clear when it will be published and ready for use.

Moreover, ML-enabled systems tend to necessitate non-functional requirements to a greater extent than non-ML based systems [54]. This is especially true for non-functional requirements such as fairness, transparency, privacy, security, and safety. Research has also pointed out that it is not yet clear whether the expertise that has been gathered in defining non-functional requirements for traditional systems can be applied to ML-based systems [54]. These requirements can also be more difficult to define as the metrics of certain non-functional requirements can be very domain and context specific. For example, fairness has been identified as a crucial, yet difficult to define, requirement of ML-enabled systems [25, 23].

Furthermore, research has also found that quantifying non-functional requirements for perception systems is especially challenging [12, 59]. Moreover, safety has traditionally been a key requirement in the automotive industry. However, with the introduction of ML, defining and communicating safety requirements for automotive systems has proven more difficult [60]. This again is due to the complexity and non-deterministic nature of ML algorithms, which can lead to difficulties in specifying and verifying safety-critical behaviors [61].

3.3 RE for multi-party collaboration

Developing a complex data-intensive product often involves a collaboration of multi-disciplinary teams. These teams can be made of different professions, from engineers and designers to auditors [62]. Various stakeholders may have different interpretations and priorities regarding safety requirements, making the development process challenging [51].

Companies focusing on creating safety-critical systems are developing agile methodologies for their workflow [63, 64, 65]. Transforming these companies, which are characterized by slower development time, into a more fast-paced environment, brings out some issues related to collaboration and requirements engineering in non-deterministic systems [21, 37].

Nahar et al. [62] examines a few communication problems when developing ML systems. First, expectations on the system might be unrealistic. Therefore, teams need to understand the whole complex system and communicate what is possible. They also found that requirements are often written and communicated informally, making updates and later clarification difficult. Dey et al. [66] also found that one of the main challenges of ML-enabled systems development is the lack of collaboration between experts. However, when more people and teams are present, the evaluation of the system becomes more difficult because of slower communication [12, 21]. This makes receiving feedback and synchronization take a longer time. This is especially true with safety-critical perception systems as developing different sensors and hardware can be a time-consuming process. Proposed solutions to speed up the communication include better requirements visualizations and adding domain experts or product managers to teams [37, 21, 62].

Kasauli et el [21] investigated several challenges related to the employment of efficient communication practices from requirement representations. Teams often adhere to their established practices and tools, making it difficult to gather data across different teams. They also claim that using different tools may result in extra effort being needed to integrate external requirements into their own systems. Furthermore, a lack of knowledge regarding the other teams requirements may lead to misinterpretation of them, resulting in incomplete or wrong requirements.

Representing requirements is challenging due to teams using different processes and methods [21, 8]. Methods for requirements representation can differ, making the requirements communication difficult between teams, suppliers, and stakeholders. This is due to everyone being used to their own methods. Natural language and modeling, using formal requirements [62] or reusing requirements [8] could ensure a more efficient solution.

Synchronization of information is challenging, due to safety-critical systems being developed in a multi-party context. Communicating these changes and making relevant modifications across different teams can take too long or be expensive. In a system where there are many dependencies, even small changes may require updating many parts of the system [8, 35]. The best option is to define safety requirements in the beginning to ensure a safe system [38]. However, since the system is iteratively developed, requirements may change, making it difficult to set a clear goal [8].

4
Methodology

This chapter describes the research methods used in the study. The chapter begins with a description of the used methodology and then detailed descriptions of each research stage.

4.1 Field study

In software engineering, a field study is any research conducted in a specific, real-world setting to study a specific software engineering phenomenon [67]. It will allow for unobtrusive research and help to develop a realistic understanding of the problem in case companies. A method for conducting a field study is a case study. In this thesis, through an interview study, data is gathered and the outcome of the case study is evaluated through a workshop and a survey.

Case studies investigate a situation in depth and within its real-world context [68]. They allow for gaining a deeper understanding of how and why some phenomena occur and can reveal mechanisms by which cause-effect relationships occur [69]. This makes it a suitable option for the thesis as we will be able to observe, gain understanding, and collect data about processes and challenges.

Runeson et al. [70] describes five major steps for a case study. These are defining the goal, preparation for data collection, data collection, analysis, and reporting. However, the case study methodology design is flexible and can consist of numerous iterations over the steps. Therefore, the plan for our research consists of these stages: preparation for data collection, data collection, data analysis, evaluation, and reporting. The plan is visualized in Figure 4.1.

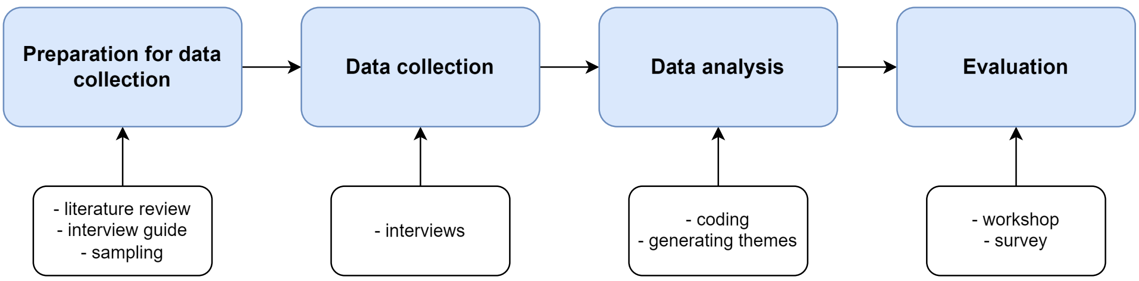

Figure 4.1: Overview of stages for methodology

Preparation of data collection consists of a literature review, designing the interview guide, and creating a sampling strategy. In the data collection, interviews are conducted. After that, the data analysis is done on the qualitative data using labeling and coding. These findings are then evaluated in the evaluation stage by attending a workshop and conducting a survey. In the end, results are reported.

4.2 Preparation for data collection

Several actions were taken to prepare for the data collection. First, a literature review was conducted to gain a deeper understanding of current problems and practices in the industry. An interview guide was designed based on the findings of the literature review. Finally, participants were sampled for the interviews.

Interview guide

The interview guide is a document containing interview questions and additional necessary information that was to be given to the participant. Its purpose was to ensure no questions were forgotten and that every interview was conducted uniformly.

Strategies proposed by McNamara [71] were followed when designing the guide and creating interview questions. They proposed open-ended, neutral, clearly worded questions one at a time and refrained from asking unnecessary questions to keep the topic on point. With this approach, questions were formulated based on findings in already existing research.

In the end, the interview guide consisted of several topics. Firstly, the participant was informed of what to do if they had any questions or if something broke during the interview. Then, general questions about the participant and their company were asked, and terms and concepts were established. This was to ensure that they had an understanding of words that could be interpreted differently during the interview.

The next topic was about requirements representation in perception systems, which included questions about representing functional and non-functional requirements. This was followed by asking about processes, what works well, and challenges in multi-party collaboration. Moreover, questions related to ML-enabled systems were asked. This includes questions about data integration, processes, and challenges in requirements representations. Lastly, potential solutions were introduced and participant's opinion was gathered. In the end, wrap-up questions were asked, and the interview was concluded. The final interview guide document can be found in Appendix A.

Questions were iteratively revisited and improved slightly, if necessary, after interviews. These improvements were small and carried out in a way that the interview would stay the same later. For example, when defining fairness requirements term, participants had an issue with the original definition. We introduced additional definitions taken from another paper and asked the participants to choose which was more suitable.

Sampling

According to Shull et al. [72], case studies should use purposeful sampling rather than random sampling. Participants were selected so they could give their unique and critical insights into the processes of case companies.

The aim was to select participants working mostly with perception systems, having a couple of them from every company involved in the supply chain. They should be working in management or development, have arbitrary experience with developing the software, and should either facilitate or receive requirements.

To select people, a document containing short information about the study was sent. This included the topic description and asking to conduct the interview.

Four distinct organizations participated in the study. Three of these organizations form a supply chain in the automotive industry: an OEM, a Tier 1 supplier, and a Tier 2 supplier. The fourth organization was a research institute.

In the end, ten people were interviewed. Every person added different experiences with the development of a perception system and works in different positions. Their company nor supply chain position is not revealed to ensure that the interview quotes cannot be traced back to individual persons. The overview of participant information can be seen in the Table 4.1.

ID	role	years of experience	I	R
I1	System designer	6+	✓	
I2	Scrum master	6+	✓	
I3	Senior safety engineer	8+	✓	
I41	Technical specialist	10+	✓	
I42	Senior system engineer	20+	✓	
I5	Researcher	couple of years		✓
I6	Perception system	10+	✓	
I7	Senior researcher	10+		✓
I8	System design engineer	2+	✓	
I9	Data scientist	2+	✓	

Table 4.1: Participant's role and experience (I - participant is from industry, R - participant is from a research institute)

4.3 Data collection

This section describes the process of data collection where interviews were conducted.

Interviews

The purpose of the interviews was to gain a deeper understanding of practices, challenges, and strategies in requirements representation and multi-party collaboration in the development of safety-critical systems. Each interview took up to an hour. Interviews were conducted over Microsoft Teams. Each interview was recorded and transcribed. For the transcript, the use of an automatic transcription tool provided by Microsoft Team was used. However, the tools were not always reliable which is why we conducted manual checks of the recording and fixed errors in the transcriptions. Collected data were securely stored and anonymized. During the interview, one interviewee asked the questions and an additional researcher took notes and observed. Questions for the interview were taken from the interview guide. These questions could be slightly altered during the interview to be better tailored to the participant's domain and expertise.

At the beginning of the interview, the participant was asked for consent to participate, be recorded, and was reminded about the content of the study. Then questions regarding their role and company were asked. This was followed by establishing terms and concepts, which were to ensure a common understanding of terms used in the interview. For every term, a definition was provided. The participant either agreed or had comments on the definition. This sometimes started a short discussion about the concept, and a common ground was agreed upon.

In the next step of the interview, questions about processes were asked. The first section was about representing functional and non-functional requirements. In this part, we identified different strategies for representing requirements and using non-functional requirements in their job. Second, questions about multi-party collaboration were asked. The questions targeted challenges, practices, and what works well when representing requirements within their own company and also when communicating outside the company. Then, questions about ML-enabled systems followed. These questions asked about data integration, challenges, and requirements representation that came with the introduction of ML to traditional systems. Additionally, questions about considering fairness requirements when developing safety-critical systems were asked. The final questions addressed participants' opinions about using shared language and reference architecture. To wrap up the interview, we asked if the participants thought some topics were missing and asked about additional persons for the interview.

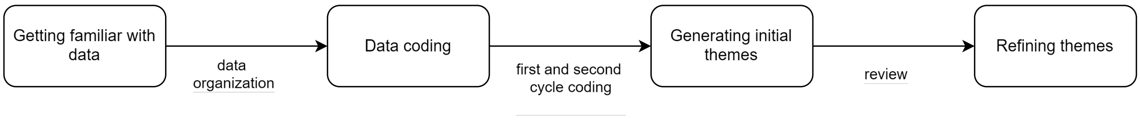

Figure 4.2: Stages of data analysis

4.4 Data analysis

This section describes the process of data analysis for the study. The stages of data analysis are shown in Figure 4.2.

The first step was to get **familiar with the data**. This was done by manually revising automatic transcripts of data, and copying them in an Excel file. The second step was **coding the data**. For coding, approaches described by Saldaña [73] were used.

After the coding, **initial themes** were generated. First, sub-themes were identified through thematic analysis. Sub-themes were then merged into main themes. Themes were **reviewed and defined** in the form of a mind map. Results are shown in Appendix C. During a brainstorming session, themes were finalized by **refining** and proper defining and naming. The final figure is shown in Appendix D.

Data coding

Saldaña [73] suggests conducting coding analysis using a "generic" method for the start. He listed several basic coding methods to start with and recommended to be open to change if necessary. Coding methods are divided into two parts - ones that should be done in the first cycle coding which is followed by methods that should be done in the second cycle coding.

First cycle coding

For the first cycle coding, Saldaña recommends attribute, structural, descriptive, and in-vivo coding. For the start, we selected the first interview to do the pilot coding. We checked whether the proposed approach provided enough information to us. After trying it, these coding strategies proved to be useful and were used later on.

Attribute coding provides basic descriptive information such as demographics and participant characteristics [73, Ch. 3]. In the data analysis, it helped with coding mainly information about the company and the role of the participant. Structural coding applies a phrase to label interview data based on the concepts of the question [73, Ch. 3]. Its usage was for categorizing the interviewee's answer to frame the information and was used the most. Descriptive coding summarizes the basic idea of the answer in a short phrase [73, Ch. 3]. It was used to summarize the statements of the interviewee. In-vivo coding refers to the specific phrase in the answer [73, Ch.

3]. It highlighted key and relevant concepts from the participant's answers.

Second cycle coding

Saldaña describes second cycle coding as an advanced way of reorganizing and reanalyzing gathered and coded data. This means linking and merging codes and removing codes that will not be used in the end. The goal is to develop a categorical and thematic organization of codes for thematic analysis.

For this part, two coding approaches were chosen - pattern and focus coding. Pattern coding groups together similar codes and summarizes them into a smaller set of themes [73, Ch. 5]. It was used to come up with coding representing similar data. Focused coding is used to develop categories [73, Ch. 5]. Focused coding was used for organizing coding based on similar categories. Together with the pattern coding, a list of sub-themes and categories was created. It is shown in Appendix B.

4.5 Evaluation

This section presents two approaches taken for the evaluation of interview results. First, a brainstorming workshop regarding the shared language was observed. Finally, our results were evaluated based on a survey.

Workshop

To connect the thesis results with the FAMER project, a workshop was observed. This also encompassed a first step in the evaluation of our results.

The aim of the workshop was to get familiar with the shared language from WP1. However, at the time of the workshop, the shared language was not finished. We were still able to observe participants' discussions on issues that occurred which made the creation of it delayed.

The workshop took two hours and participants had introduced their progress regarding shared language, which was observed by us. The structure of the workshop was mainly to brainstorm and discuss the progress. We listened to their presentation and took notes.

Survey

To validate the relevancy of our findings, a survey was created and sent to people who participated in the interviews. The survey was a good option on how to evaluate all our findings as it allows for a quick gathering of responses that can then give another insight into our findings.

The role of the survey was to present the identified approaches that work well and challenges to people who participated in the interview and ask them to which extent they agreed with the results. In addition, we explored the influence of the shared language and reference architecture on identified challenges and approaches that work well.

The focus of the survey was to get the participant's opinion on:

1. to which extent the participant considers the identified practices to be effective,
2. to which extent the participant considers the identified challenges to be relevant,
3. whether the shared language and reference architecture may break identified approaches that work well,
4. whether the shared language and reference architecture can help to alleviate identified challenges.

The survey was designed in Microsoft Forms and was divided into 5 sections. In the first part, a question about the participant company's position in the supply chain was asked. The next section presented identified approaches that work well and the participant can select on a Likert scale to which extent they agree with it. In the next section, the participant assessed whether the concepts can be broken by introducing either reference architecture, shared language, or both, or if neither can break it.

The next section focused on identified challenges. First, identified challenges were presented, and participants can mark on the Likert scale to which extent they agree that the identified challenge is relevant. Second, participants were asked whether the challenges could be alleviated by introducing either, reference architecture, shared language, or both, or if neither can help to alleviate the challenge. In the end, there was an open question where the participant could describe their thoughts to some answers if they wanted.

The link for the survey was then sent out to people who participated in the interview by email. To encourage people to fill out the survey, we kept the questions short and precise. The expected time to complete the survey was ten minutes. In the end, we received answers from three people.

5
Results

This chapter presents the results of the thesis. First, the findings related to RQ1 and RQ2 are presented. The findings are divided into three distinct categories: identified current practices, approaches that work well (RQ1) and identified challenges (RQ2). All identified themes related to RQ1 and RQ2 can be viewed in Appendix D Finally, the findings related to RQ3 are presented. When presenting the results we are considering three main themes. These are requirements representation, multi-party collaboration, and RE for ML, see Figure 5.1

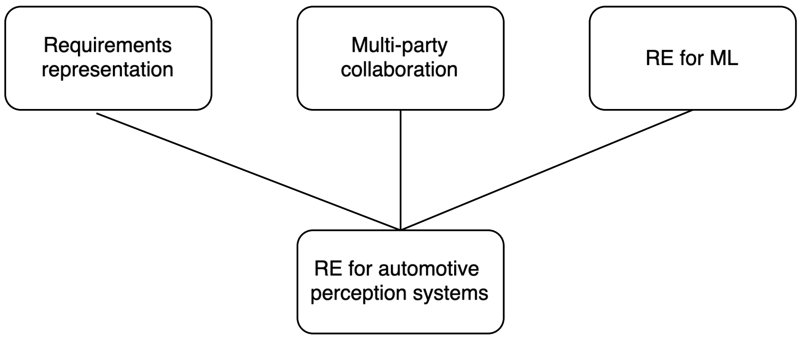

Figure 5.1: Three main themes of RE for automotive perception systems

5.1 Current practices

In this section the current practices of the case study companies are presented.

Requirements representation

The current practices of requirement representation are visualized in Figure 5.2. To represent requirements for perception systems, the interviewees stated that they use natural language. ODD's are used and defined as a way to limit the problem space and describe the environment where the system will be active. As such, ODD can be classified as a way to represent requirements. Furthermore, annotation companies tend to use key performance indicators (KPI) as a way to represent requirements. Moreover, images are used as a way to further visualize certain aspects of desired system behavior.

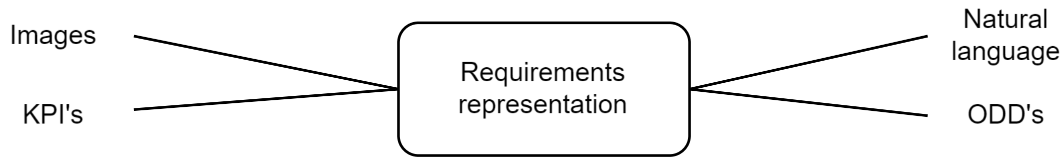

Figure 5.2: Current methods used for requirement representation by the case study companies

Non-functional requirements

When working with safety-critical systems, participants mentioned that they mainly consider two non-functional requirements - safety and performance. While safety requirements should ensure the development of a safe system, performance requirements are also complementing safety by focusing on correct and timely responses.

Other mentioned non-functional requirements included fault tolerance, interoperability, accuracy and privacy. Capacity and latency were also mentioned, especially concerning the networking capacities of cars. Additional mentions included computer architecture, standards for communications, execution policies, and deployment procedures.

Multi-party collaboration

The five main themes of multi-party collaboration can be seen in Figure 5.3. For multi-party collaboration, the interviewees explained that they tend to rely on a more formal communication process compared to in-company collaboration. This includes having dedicated communication channels, communication hierarchies and sync-meetings. Furthermore, interviewees described a need-to-know based view on collaboration when working with suppliers further down the supply chain. In practice, this means that certain contexts and system designs of the OEM may not be shared with the supplier if it is not deemed critical to do so.

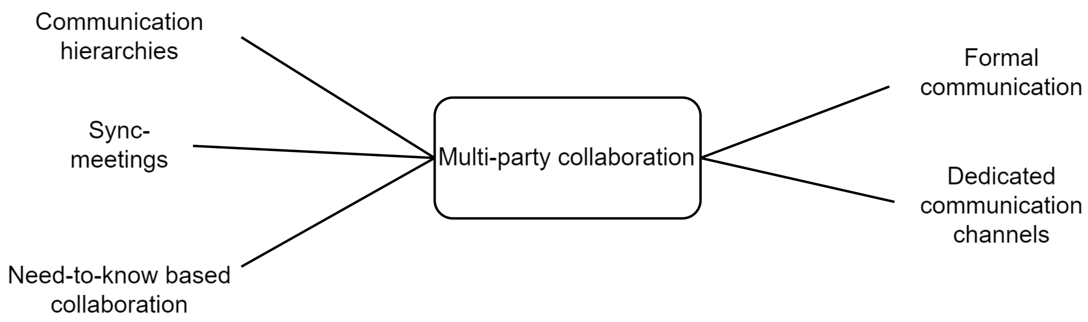

Figure 5.3: Identified themes of multi-party collaboration for the case study companies

ML requirements

For requirements related to ML, the interviewees stated that they tend to use percentage thresholds to define a required accuracy for a ML-model, see Figure 5.4. For annotation requirements, the interviewees explained that it is mostly defined by KPI's and annotation guidelines given by the client. Participants also pointed out that there is a lack of industry standards available for the development of ML-enabled vehicles as compared to traditional vehicles. Furthermore, when it comes to defining and working with fairness requirements, the interviewees state that they tend to not define fairness requirements explicitly. However, participants pointed out that fairness related concepts are included implicitly in performance and safety requirements.

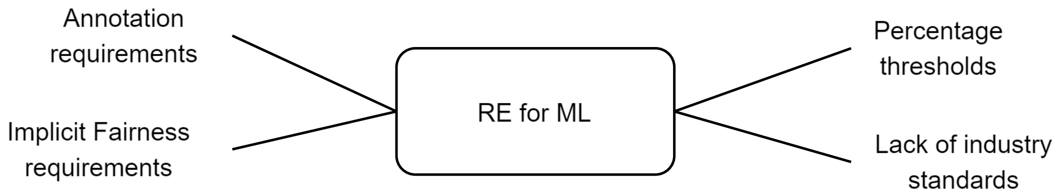

Figure 5.4: Identified themes of RE-approaches for ML used by the case study companies

5.2 RQ1: What approaches work well?

In this section the approaches that the participants felt worked well are presented. Figure 5.5 shows the main identified approaches that work well. Further explanation will be given in text throughout the section.

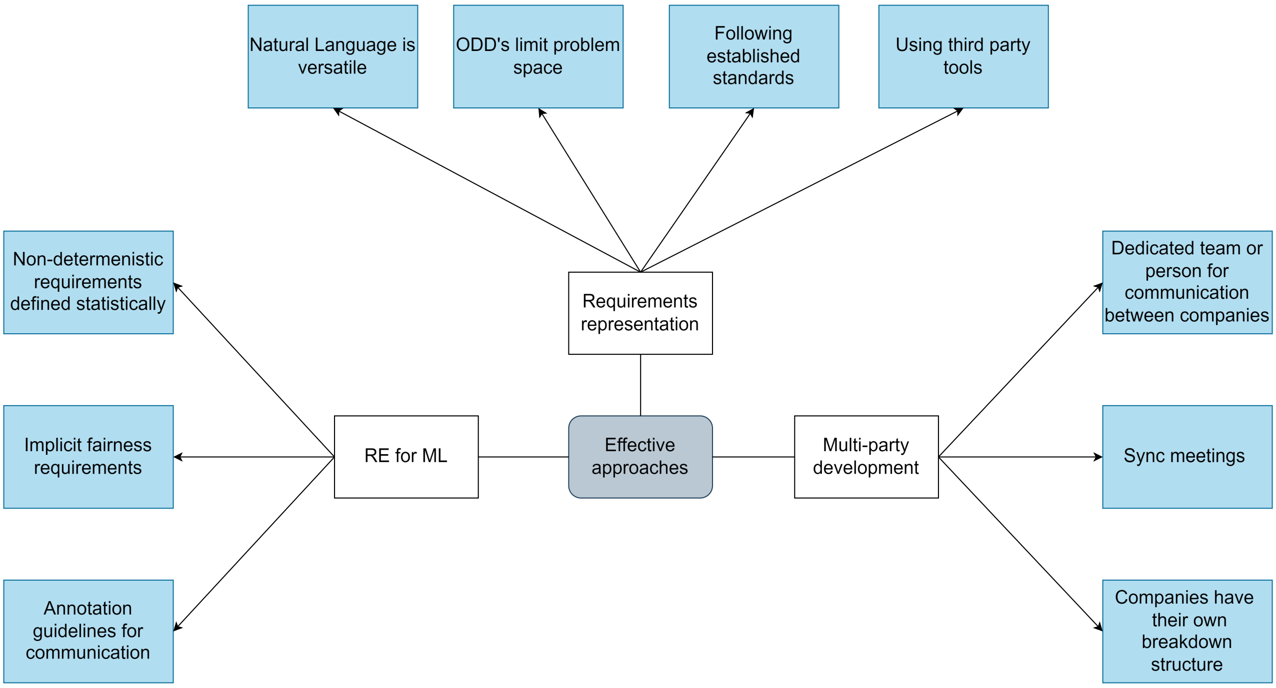

Figure 5.5: Identified effective approaches for RE for automotive perception systems

Natural language is versatile

Each participant mentioned that the requirements representation is done using natural language. This can be done using some tools, pseudocodes, or just PDFs with text and images.

Representing requirements using natural language ensures some versatility. As safety critical systems depend on different nonfunctional requirements, it is easy to represent different constraints of requirements. In this form, they can be broken down as far as necessary or can be sufficient on a high level.

Following established standards

Participants generally follow established standards. Each participant mentioned following *ISO 26262* [29] during their work. Other mentioned standards were *ISO 15288* [74] and *ISO 214343* [75]. Additionally, some standards and guidelines can be set internally by a company in the supply chain that understands a specific system the best.

ODDs limit problem space

ODDs can be useful for specifying the domain. They provide enough information about the system to break down a complex structure into more feasible parts. This can provide an overview of system's possibilities and limits. Developers can focus on their work while only knowing enough about their domain and making the system's overall information more manageable.

I41: *"They [developers] should not be unaware of the higher level, but they should be very knowledgeable in their constraints, so working on a particular component they are very clear expectations of that component."*

Using third party tools

Participants highlighted the benefits of using third-party tools in product development, particularly for documentation and management purposes. A classic requirements and systems engineering tool was used for communication between companies, while a text- and git-based tool was used internally by development teams. Participants also mentioned using Excel for requirements management.

On the other hand, it seems that for research the tools are not so required.

I7: *"I think in a research project scope this[sending documents] works fine. You don't need like requirement engineering tools or any thing that is complicated because you're not developing a kind of a product or anything. So it works fine. "*

Companies have their own breakdown structure

As the participants work with complex systems and each participant has experience on different layers, having own breakdown structure for each company is beneficial. Each team can adjust the breakdown of the used component to their abstraction level and assign it to their structures. This allows for more customizable communication within the company and outside the company.

I41: *"We have one breakdown structure, which we apply internally and some of those things are important to communicate externally because they are the boundary of the responsibility."*

Annotation guidelines for communication

Annotation guidelines are a way to represent requirements when communicating with annotation companies. Annotation companies are responsible for annotating data based on these provided guidelines. They are a less formal way to represent requirements and it makes it possible to poke around and ask for requirements information initially.

I2: *"And I guess one could see the guidelines as requirements also in the way like you have done it this way with this kind of and statistical. "*

If necessary, they can also break it down according to their needs.

I6: *"...we are breaking those requirement from the client to put on our tooling or like later in engineering for example if they need a specific feature and that's requested by the client "*

Dedicated team or person for communication between companies

Several participants mentioned having a dedicated person or team responsible for communication between companies. This allows the developers to focus mainly on development. However, developers can have their own channels for quick clarifications between teams if necessary.

Sync meetings

Having common meetings when people are caught up with recent updates about the project is also beneficial. These meetings can be formal or informal, where the information is communicated using simple working tools and structures (for example Excel tables).

I8: *"The informal could be we just have a meeting with them saying, OK, so your component which you develop, we have three new requirements. So let's just go through them in a common meeting and get an understanding. Uh to avoid multiple flows, basically. "*

Non-deterministic requirements defined statistically

As most ML systems are black boxes, it is difficult to interpret certain requirements for the system.

I5: *"..essentially you have this black box which you can't reason about. You can't argue about it, so there is a lot of challenge [...] you have to have representative data sets and they have to be correct and they have to be fault-free. And how can you guarantee that?"*

To deal with the black box characteristics of ML, some participants mentioned that the best approach to tackle this is to use a statistical approach. That means expressing requirements using percentages or intervals, they should be well-defined in constrains and should be approached as a math problem.

I41: *But you can have different ways of analyzing outcomes. Like you can have like a mathematical problem. There's more than one solution, right? And that's fine. You can have more than one solution, but the solution needs to have an accurate* *boundary.*

The verification of different metrics can check whether the system fulfils a required KPI. KPIs are also a way of representing requirements for the annotation companies as it make it easy to see what thresholds or intervals need to be achieved.

Other mentions on how participants are handling non-deterministic properties include using ODDs and other requirements (for example performance).

Implicit fairness requirements

Participants stated that even though they do not explicitly define fairness requirements, fairness related concepts are implicitly included in performance and/or safety requirements. The interviewees argue that if a system is worse at detecting a certain type of pedestrian, no matter the reason, then it is by definition a less safe system. Moreover, interviewees pointed out that different types of pedestrians are considered in requirements. However, interviewees defined this as related to performance rather than fairness. One interviewee stated:

I8: *"[...] the person on a wheelchair has a smaller area compared to a person standing and kids occupy a smaller area compared to adults. So we define what a person is, min and max, you could say. [...] So we don't classify male or female, we just leave it to a person basically."*

When asked about the classification about these requirements the interviewee stated that it would be classified as a performance requirement.

Participants also claimed that the underlying reason for potential fairness related problems was the same as the underlying reason for most problems with ML-based predictions. Namely, a lack of representative data.

I1: *"I mean, it's not a fairness requirement if the vehicle stops every time a road sign is tilted 4.75 degrees. But it's still the same underlying reason that we maybe didn't gather the representative data."*

5.3 Validation results of effective approaches

This section presents results related to RQ1 from the validation survey. Identified approaches that work well were shown to the participants and they could express to which extent they agree with them. Results are presented in Figure 5.6.

Participants mostly agree with the identified effective approaches. However, there is disagreement regarding implicit fairness criteria and varying views on KPI's representation and having a dedicated person for team communication. They also have mixed opinions about the benefit of own breakdown structure.

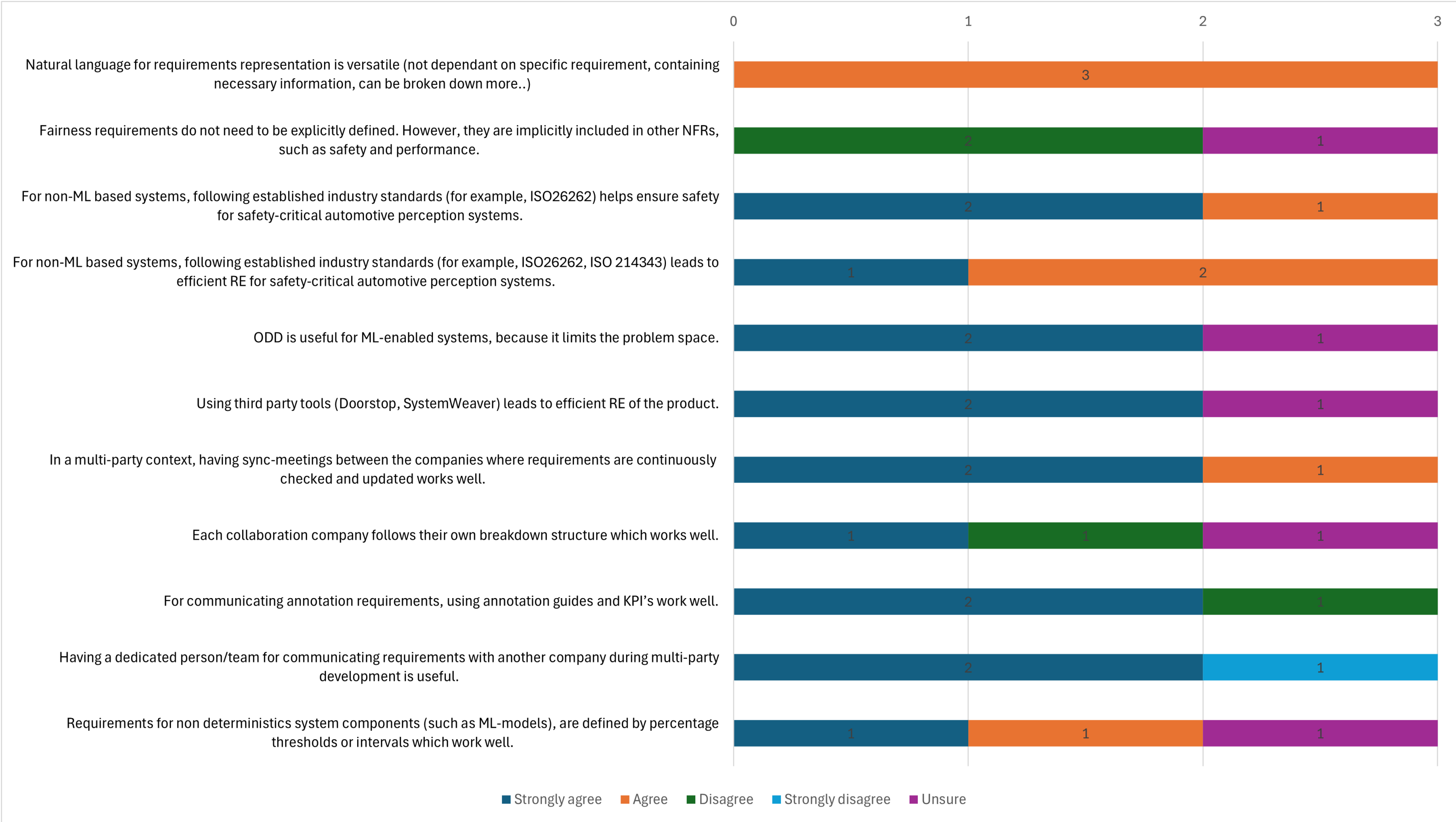

Figure 5.6: Validation survey results for RQ1

5.4 RQ2: Challenges with current approaches

In this section the identified challenges are presented. In figures 5.7 and 5.8 the seven main identified challenges are highlighted in blue. The figures display the challenges, their relationship to each other and related consequences of these main challenges. Further explanation is given in text throughout this section.

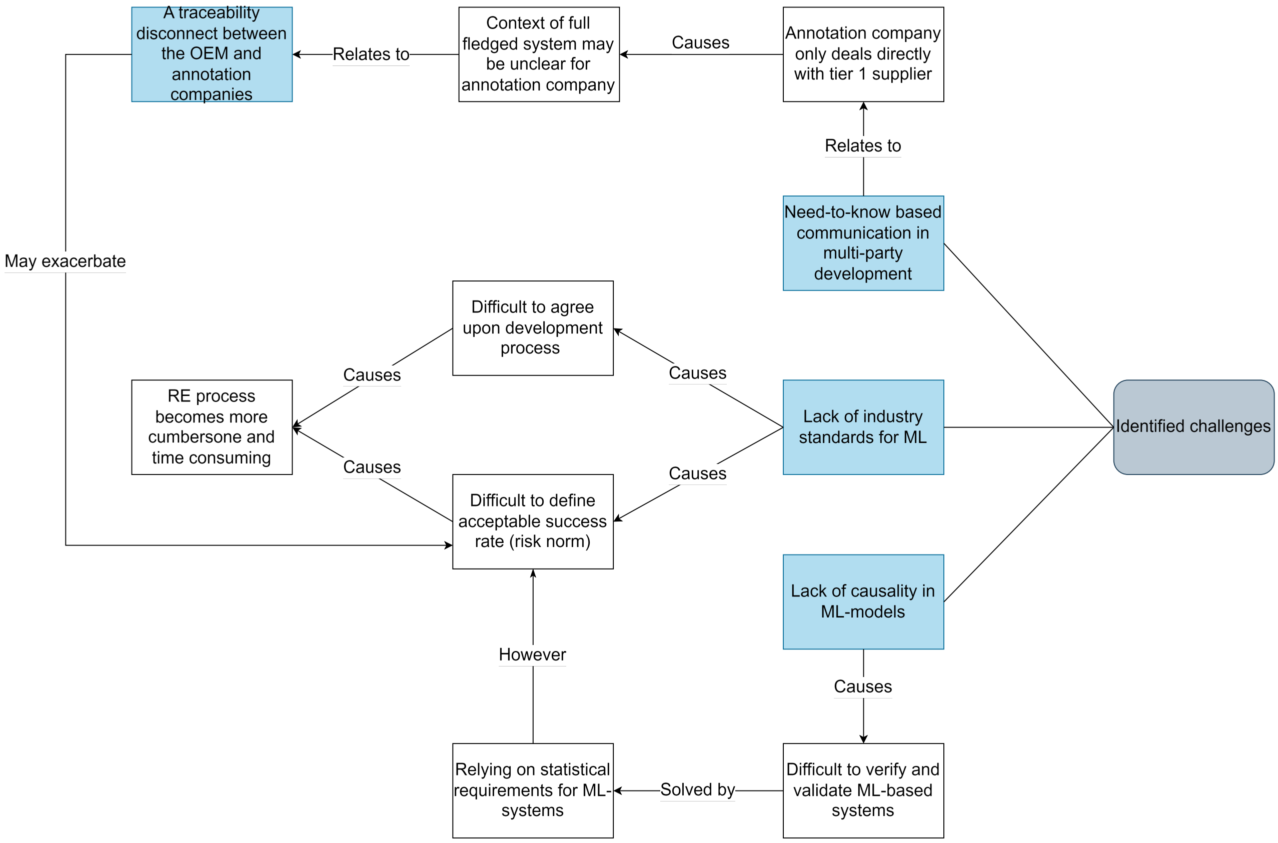

Figure 5.7: Identified RE-challenges for automotive perception systems (1)

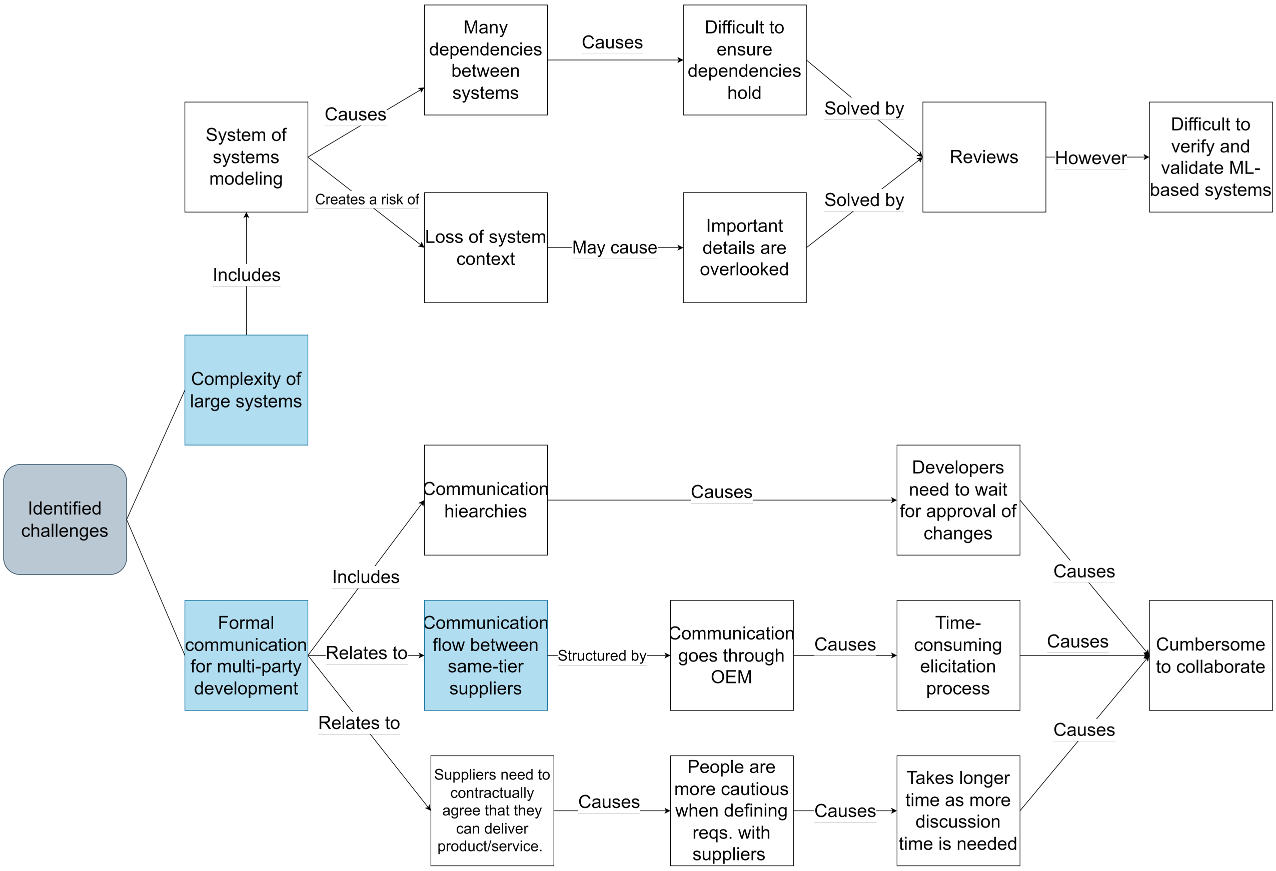

Figure 5.8: Identified RE-challenges for automotive perception systems (2)

Lack of industry standards for ML

The interviewees pointed out that RE-processes for non-ML-enabled vehicles is very standardized and has been iteratively improved upon over a long period of time. However, as the capabilities of ML has increased so rapidly over a relatively short period of time, comparable standards for ML-enabled vehicles simply don't exist yet. This leads to some challenges that will be presented below.

As previously mentioned, the case companies tend to use statistical requirements to evaluate the accuracy of ML components. But due to a lack of industry standards, the interviewees stated that there is a difficulty with defining acceptable thresholds for statistical requirements. Certain aspects may be more critical than others, but defining if a certain ML-model needs to have an accuracy of 90%, 95% or any other threshold has been identified as a difficult process.

I3: *"And especially on top, how much is OK to accept the risk? Because if you reduce the percentage then you might face some risks. But how much is good enough?"*

I5: *"Yeah, there is this. What level of risk in society do we accept? This, risk norm"*

Moreover, interviewees explain that the lack of industry standards leads to the process of defining suitable requirements taking longer time compared to traditional systems. In a multi-party development context, there may be opposing views on how to best solve or approach a particular problem. This is of course not unique to ML. However, in traditional systems companies could reference to industry standards and argue why a certain approach is preferable or maybe even strictly required. As such regulation and standardization does not exist yet for ML, it is challenging for collaborators to argue why a certain process is superior to another.

Lack of causality in ML-models

Due to the black box nature of ML-models, the interviewees stated that it is difficult to argue for the safety of ML-enabled systems. Since it is difficult and in some cases even impossible to explain why a ML-model decided to take some particular action, or fail/succeed to identify a certain object, it is hard to reason or argue about the safety of the vehicle. The current practice is to use statistical metrics to measure the accuracy of the ML-models. However, as mentioned in the previous section, this brings on challenges of its own.

I5: *"There is no causality, there is no repeatability. It [ML] can solve problems, but if you can't explain why it can, it's kind of useless for a lot of applications that are safety critical."*

Furthermore, interviewees stated that this lack of causality means that it becomes difficult to verify & validate ML-requirements. Depending on what percentage threshold has been determined to be sufficient, it may require a large amount of verification data. Interviewees point out that a ML-based prediction that requires an accuracy of for example 90% is easier to statistically verify as compared to a required accuracy of 99,999%.

Formal communication for multi-party development

Interviewees pointed out that in multi-party development, the communication process tends to be more formal as compared to in-company collaboration. This often leads to a more time-consuming process when trying to define and agree on suitable requirements. Furthermore, if changes occur during development and certain requirements are deemed unrealistic or to abstract, interviewees point out it becomes cumbersome to communicate these changes. Often there are certain communication hierarchies that these changes need to go through in order to get approved.

Communication flow between same-tier suppliers

It was stated that the OEM may have multiple Tier 1 suppliers that supply software and hardware for the same subsystem. Interviewees explained that in such cases the OEM acts as hub for all communication. For example, a Tier 1 supplier (supplier A) may be responsible for supplying some software solution. They would then communicate the hardware requirements that must be fulfilled in order for that software to perform to the OEM. The OEM would then contact a different Tier 1 supplier, (supplier B) and present them with the requirements given by supplier A. Interviewees stated that usually there would then be a common meeting where all three parties meet and discuss feasibility and potential issues. There could potentially be multiple such meetings in order to completely agree on the requirements. This process was raised as being very time consuming and cumbersome.

I8: *"The OEM exports those requirements to the supplier and then in a common meeting we go through. OK, these are the new requirements. Does the supplier understand? Is it possible or not? We have like a huge Excel sheet with different status accepted, not accepted. Discussion needed, rejected and so on. It's super cumbersome"*

Need-to-know based communication in multi-party development

When collaborating with outside companies the interviewees point out that the further down in the supply chain a company is, the less context and information about the main system will be shared. Interviewees point out that this can potentially lead to engineers lacking necessary context that may be needed to fully understand a systems needs.

Traceability challenges going from top-level to annotation requirements

Related to the previous challenge of need-to-know based communication, interviewees claim that there is a lack of traceability when going from top-level requirements to annotation requirements. This is due to the fact that an annotation company only deals with the Tier 1 supplier, who then communicates with the OEM.

I6: *"We cannot find any traceability between like how we are communicating to our client and how our client is communicating then to their clients, which typically could be eventually an OEM. That's also something missing and a bit like mysterious"*.

However, it should be stated that the severity of this challenge may not be very high. As one interviewee points out that no clients have specifically brought this issue up during collaboration:

I6: *"I'm saying that like there is no traceability between what an OEM wants until it comes to the data annotation to us, but I'm not convinced that if the traceability is provided, how it can help. Maybe for some safety argumentation that could be used, but up to now we have not seen this need from our clients."*

Complexity of large systems

Interviewees pointed out that when working with large systems it is common to model them as system of systems. Meaning you have multiple systems working together to create one coherent system. In such a system of systems context, interviewees have mentioned that there can be many dependencies between these subsystems. Keeping track of and ensuring the dependencies do not break has been identified as a challenge. Furthermore, interviewees elaborate that even though they have practices put in place to prevent this, in their experience there is always a risk that things falls between the chairs.

5.5 Validation of results for RQ2

This section presents the results related to RQ2 from the validation survey. Figure 5.9 shows the results of the survey. The identified challenges related to the lack of industry standards, were mostly confirmed. With the exception of one participant disagreeing with the fact that there were a lack of industry standards. Moreover, most challenges related to multi-party development were confirmed by the participants. The only exception is the need-to-know basis which one participant marked as unsure. Most participants disagreed with the lack of traceability going from top level safety requirements to annotation guidelines/KPI's.

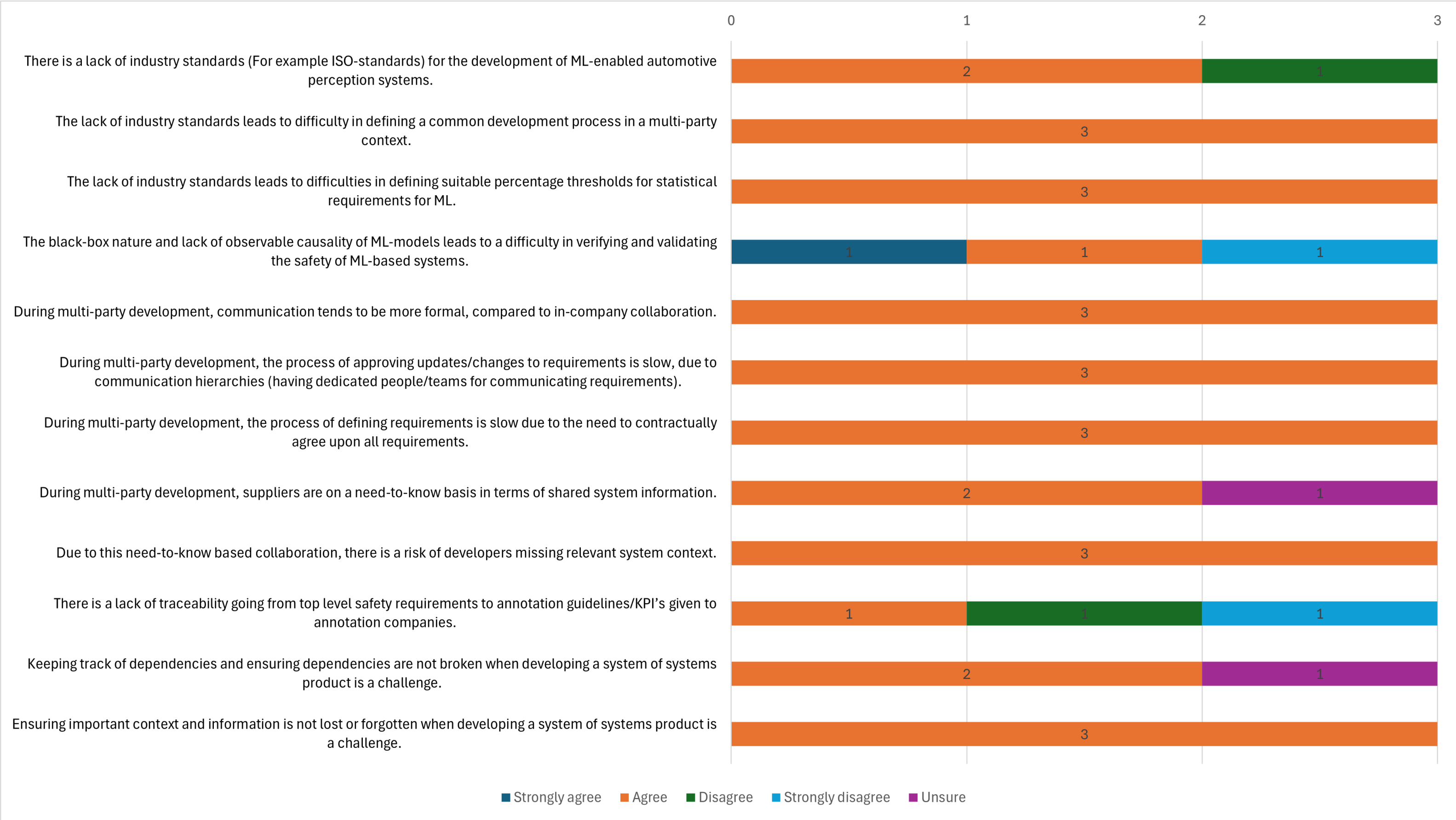

Figure 5.9: Validation survey results for RQ2

5.6 RQ3: Shared language

In this section the results regarding the potential of a shared language to solve challenges are presented. Unfortunately, the shared language which is developed in WP1 of the FAMER project could not be finalized in time for this thesis. Therefore, the results are based on the interviewees' initial opinions about the shared language, and a observed workshop for the creation of the shared language.

Opinions from interviews on shared language

In general, participants agree that having a shared language is a good idea. The main benefit is that it would improve the interpretation of terms and concepts that can differ across different disciplines.

I2: *"...it's always like you know a barrier at the first proof of concept for a new guideline to align on all of these things."*

In addition, each company may have different requirements representation leading to confusion. Shared language could provide some standardization on this process and a consensus on working together with different representations. One participant mentioned that shared language can also resolve issues with traceability among the companies.

I3: *"...for example, the first challenge is traceability of the requirements and if the parties they define the requirements in different ways then the traceability would be a challenge. So, that's the that's one of the main factors that make it important to have consensus on the common language."*

However, some consensus on terms and concepts already exist. Some terms, especially related to ML are established and understood. People align to a common understanding of different concepts and in some cases using ODD is enough.

I7: *"I think right now we have quite a good development within the operational design domain. I think people are getting more understanding of what it is and that is something communicable and all the companies know what it is. "*

Opinions from interviews on reference architecture

Participants agree that having a reference architecture would be useful and helpful. It would improve the communication by properly describing the system and help with tailoring better requirements.

I5: *"So would it help to work with and defining requirements? Of course, there is always this tailoring you have to do. A small OEM needs to tailor once solution to fit into the into their architecture and there is always misunderstandings there.*

So requirements for sure. Misunderstanding in requirements is the most common systematic fault in in all implementations."

Participants described that it is usually OEM who comes up with reference architecture. So in the end, the reference architecture would be a product of one company and other companies may not want to follow that. One participants describes that this design should be more high level and then suppliers should choose more individual approaches for the development.

I7: *"So, I think at the very high level you have reference architecture which is pretty straightforward. But when you're going down to the lower level and I think that's the kind of each company's own design, it's individual."*

Additionally, most participants agreed that it is hard to see the reference architecture happening because companies want to keep their information private. Companies do not want to share details about their processes with others, as each has their own approach and their own unique design. They may be afraid that making this information public, would reduce their market value.

I2: *"I mean, it would be nice, but I guess also it's a lot about, you know, giving too much information to other companies so could, if they are not nice, sell it"*

On the other hand, making the architecture publicly available, may lead to cheaper development of the system.

I3: *"On the other hand, it's even better for them, because if they release their architecture then it's cheaper for their suppliers to develop those architecture. Because then it's just one architecture and one system, so that the cost of the development and testing and these kind of aspects would be reduced and the competition between the suppliers also will increase."*

One participant argued that it may not be feasible to create a reference architectures for ML-based systems. As ML-based systems have a black box nature, it may be difficult to describe the architecture of such systems.

I6: *"For a black box nature of an AI system, like if you have an AI system that as an input that has like sensors and then different layers with 1000 nodes in each layer, can we define an architecture for that giant ML component?"*

While coming up with a reference architecture for ML-enabled perception may be difficult, it could be more feasible if it was designed on a more general level. One interviewee argued that there is no significant difference between perception in an automotive context or for example in a facility security context. Another interviewee added that although their implementations may differ, they align on certain shared safety goals. Therefore, creating a reference architecture for perception systems is perhaps feasible if it is kept on a more general level.

I41: *"It wouldn't be a difference if you were in an automotive context or in a facility security context. The perception system on camera would probably be organized in the same way when it comes to artificial intelligence or ML. You need the same principles."*

I42: *So it's a different set, but they also share a common set of safety goals. So that's one example that we could potentially do."*

Workshop takeaways

This section presents key takeaways from the observed workshop regarding the creation of the shared language in WP1. Two issues stood out. First, the creation of a shared language is difficult. As it involves multiple parties coming together and describing their domain and architecture, it is hard to find a suitable representation. It is difficult to distinguish the terms and different concepts. Participants also described main challenge of this creation process as needing a common language to be able to define a common language.

The second issue is the privacy of the involved parties. A company in the supply chain does not want to share information about the architecture due to privacy reasons. This makes the creation of a shared language difficult as well.

5.7 Validation of shared language

The aim of this section is to verify that the introduction of shared language preserves identified approaches that work well and whether it can help to alleviate some of the identified challenges.

Results for validation of the shared language effect on identified approaches that work well are shown in Figure 5.10. Participants mostly expressed that neither shared language or reference architecture would break the identified approach. One exception is natural language, where reference architecture may break the concept. Another exception is following breakdown structure, where both can break the concept.

Figure 5.11 shows results for exploring whether shared language could help to alleviate identified challenges. Answers indicate that introduction of shared language and reference architecture could help with alleviating most of the identified challenges. However, in almost every answer, one response indicates that neither can help.

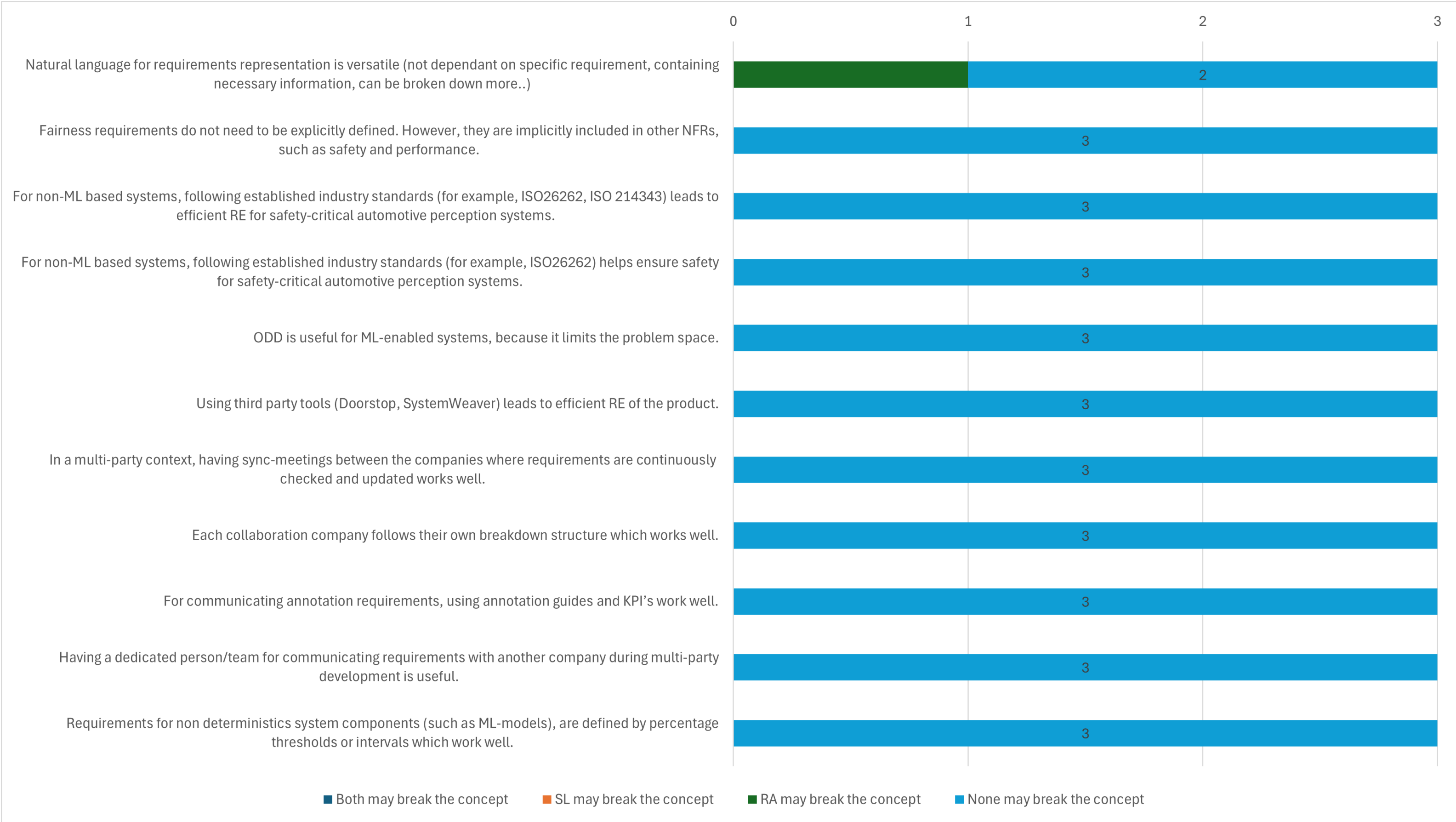

Figure 5.10: Validation survey results for the shared languages effect on processes that work well

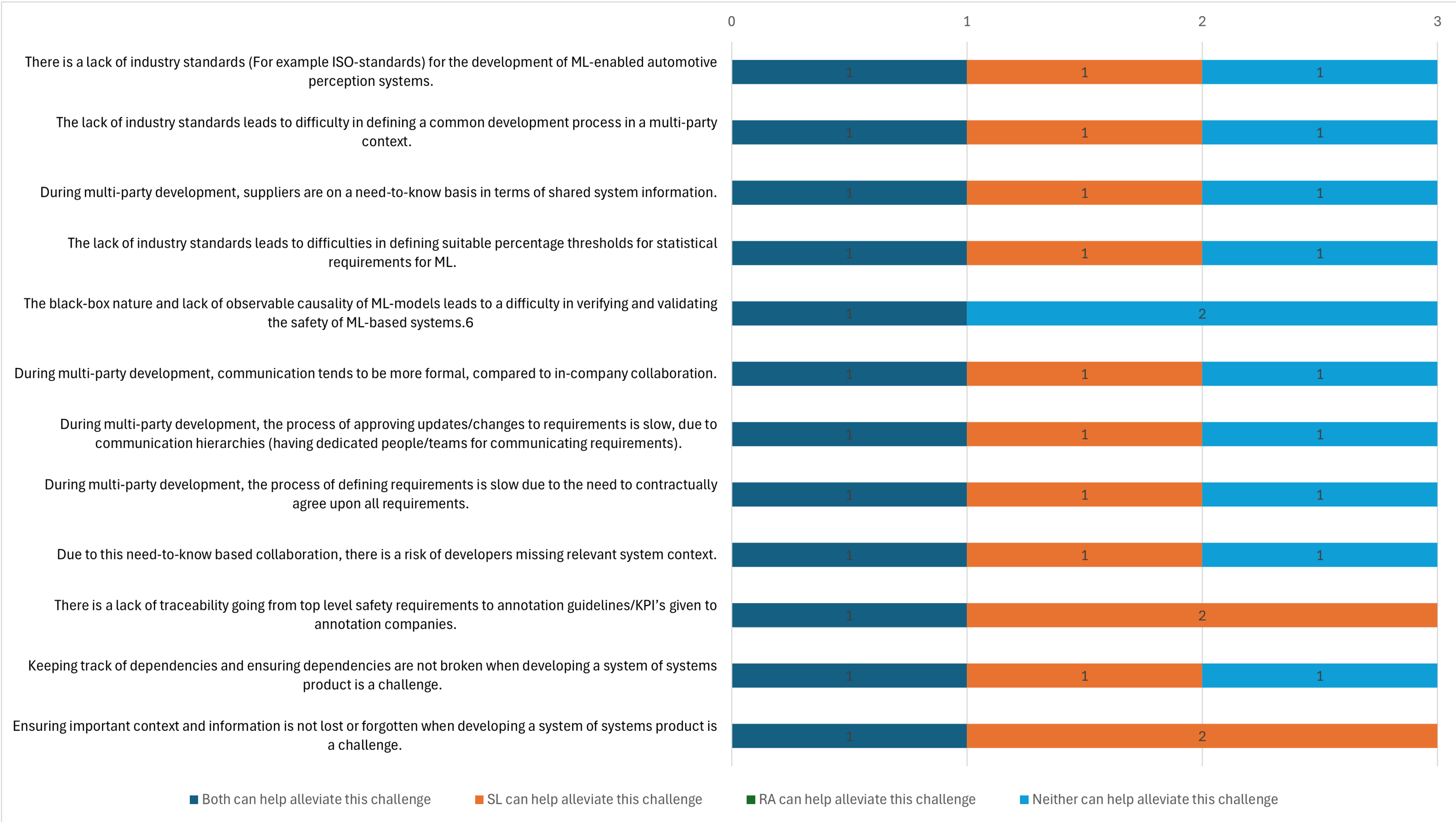

Figure 5.11: Validation survey results for the shared languages effect on challenges

6 Discussion

In this chapter the previously presented results are discussed. Furthermore, potential threats to the validity of the thesis are addressed.

6.1 Answer to RQ1: Identified approaches that work well

As an answer to RQ1, ten approaches were found to work well. Mainly, the usage of natural language is wide and versatile. The versatility of natural language can also be seen in research [53]. It is the simplest way to express requirements and it can be interpreted in needed depth. Its versatility allows the usage of different tools for representation, ranging from dedicated tools for requirements management to simple PDFs with text and images. While it may not be perfect, as there is a risk of interpreting terms and concepts differently in each company, it seems to be a sufficient way to develop ML-based systems.

Other approaches concerning requirements management aim for the effective development of a complex system. Just like research [18] suggests, following defined standards provides some pivotal structure into how to do the development. The problem space of the system can then be limited by ODDs and the developer can focus on developing while only being familiar with the product's context. This may also improve better understanding of the system's expected capabilities by the developer, which was found to be an issue by Nahar et al. [62]

For collaboration, the best approach is to have a dedicated person for communication and clarification between teams and companies. While the respondents in the survey expressed some doubts about this approach, research [35, 18, 59] proposes something similar by having a dedicated person for some discipline.

Research also suggests that one issue is the synchronization of information across multiple collaborating parties [8, 35]. While participants mentioned slower communication during multi-party development to be a problem, one of the proposed approaches is to have common meetings. In these meetings, planning is done and questions can be asked, making everyone synchronized. These meetings should be adapted to everyone's needs and pace. Annotation guides, KPIs descriptions, or other forms of less formal requirements description can also be used for efficient communication.

Research suggests that the development of a ML-enabled system is challenging because of its black box nature [14, 54, 17]. Participants generally agreed with this. However, multiple approaches exist that makes this issue more manageable. The major theme is to express requirements in the form of metrics. Expressing requirements in the form of different intervals, percentages or other metrics is sufficient, as the output can be verifiable even without understatement of what is happening in the black box. The important part is to have properly validated training data and know the capabilities of the ML-enabled system.

Many considered that the inclusion of fairness requirements makes sense for automotive systems. However, some argued that fairness related issues are implicitly included in safety and performance requirements. They argue that fulfilling fairness requirements is a subject of including diverse data in the training set.

Key findings of RQ1

- Effective ways to represent and work with requirements are using natural language, ODDs, following standards and having own breakdown structures.
- It is beneficial to have a set of tools for the representation and management of requirements.
- Having a dedicated person for communication, synchronization meetings, and annotation guidelines can improve multi-party collaboration.
- Nondeterministic requirements are expressed using metrics.
- Fairness requirements should be considered as part of other non-functional requirements and are mostly relevant for training data.

6.2 Answer to RQ2: Identified challenges

This section discusses the presented results of RQ2. The severity of the challenges are discussed as well as addressing potential connection between different challenges. Moreover, there is a discussion regarding interpretation of the results, and how the results relate to previous research.

Lack of industry standards for ML

Many interviewees pointed to the lack of industry standards as the main challenge when dealing with RE for ML-based automotive perception systems. This relates to the fact that the standard *ISO 26262* does not cover ML-concepts [57, 61]. It should be stated that an ISO-standard for ML systems exists, namely, *ISO/IEC 23053:2022 Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)* [76]. The standard is, however, not specifically for the use of ML in an automotive context. Additionally, as mentioned in the related work under Section 3.2, there is an ISO standard currently under development that aims to define standards for the use of ML in road vehicles.

Lack of causality in ML-models

The identified challenge of a lack of causality in ML-models may relate to many of the ways that RE is difficult for ML. Since for a requirement to be useful it must be verifiable, and if the system itself is hard to verify, it stands to reason that it becomes difficult to define requirements for it. Moreover, the difficulty of verifying ML-models has also been identified as a challenge in research [55, 14, 56]. It also stands to reason that if you cannot ascertain causality of a safety-critical system, it becomes difficult to argue for its safety.

Formal communication for multi-party development

Participants pointed out that during multi-party development communication tends to be more formal. Participants explained that this often leads to a slower elicitation process since things have to be agreed upon contractually. During the development of systems, it also becomes cumbersome to update and change requirements. There are communication hierarchies that must be adhered to before an update is approved, which increases the lead-time going from identifying potential issues and the resolution of them. However, it is not clear whether there is an alternative approach that is preferable. Interviewees have also pointed out that some positive trade offs exist to having a more formal approach to multi-party development. Through the interviews we observed three negative and three positive aspects of using a formal approach to communication in a multi-party context:

Positive:

1. A formal approach helps ensure clear boundaries of responsibility. This helps reduce the risk of conflict, as it is clearly stated what each party is expected to deliver for the final system.
2. Expertise is modularized which means that developers can focus on development, while product owners and system designers deal with out-of-company communication and complex RE tooling.
3. Communication hierarchies make it less cumbersome for people responsible for communicating changes/updates across parties. This is due to only having to contact one person or one team for a specific change, instead of potentially multiple teams.

Negative:

1. Elicitation may take longer, as suppliers need to contractually agree upon system requirements.
2. Expertise is modularized which means the developers may not understand how to structure requirements, which makes it hard for them to identify incorrect requirements, and effectively argue against them.
3. Communication hierarchies make it time-consuming to update and change requirements.

Communication flow between same-tier suppliers

Related to the formal nature of multi-party communication, interviewees stated that in cases where multiple Tier 1 suppliers are working together, the communication flow always goes through the OEM. This has been classified as cumbersome by interviewees. However, the alternative may not be more efficient. If a Tier 1 supplier were to communicate to another Tier 1 supplier without the OEM being in the loop, context may be lost. Since the OEM is the final customer, it is understandable that all related communication should include the OEM.

Need-to-know based communication in multi-party development

The need-to-know based communication has been identified as challenge, as companies further down the supply chain may have limited context regarding the developed system. However, in certain cases suppliers may only need a small amount of system context to be able to deliver their intended product. As such, sharing more information regarding the complete system may have little to no positive effect. However, all participants of the validation survey agree that there is a risk of developers missing relevant system context due to this need-to-know based communication, see Figure 5.9.

Traceability challenges going from top-level to annotation requirements

It could be argued that traceability is one of the main reasons to have requirements. This is especially true in safety-critical systems, where having a safety case often is required by safety standards [77, 78]. Having clearly traceable requirements of the system aides in safety argumentation for such a safety case. Even so, it has been identified that the traceability from top-level requirements down to annotation guidelines is a challenge. However, two of the three participants in the validation survey disagree with the claim, see Figure 5.9. One of which even strongly disagrees. Therefore, it is not clear whether this is in fact a challenge for the case study companies.

Complexity of large systems

The challenges related to the complexity of large systems may be more generic and not a problem specific to automotive perception systems. Furthermore, this challenge is mainly solved through reviews and these reviews have been identified as being quite effective. However, interviewees have pointed out that reasoning about ML-based systems is more difficult as compared to traditional systems. Therefore, reviews may be less effective and more difficult in a ML-context. This may then potentially exacerbate this challenge when dealing with ML-related components.

Key findings of RQ2

- The lack of industry standards for ML-enabled automotive perception systems makes it difficult to collaborate in a multi-party context.
- The lack of causality in ML-models makes it difficult to define requirements for ML-enabled automotive perception systems.
- The formal approach to multi-party collaboration leads to a more time-consuming and cumbersome RE process.

6.3 Answer to RQ3: Introduction of shared language

Overall, participants agreed that the introduction of shared language and reference architecture is a good idea. It would simplify the development processes, improve tailoring requirements, and provide solutions to some extent to identified challenges.

The main thing seems to be that shared language would bring a bit more standardization into the system development. This is also the case in the research where different visualisations [21] and more precise natural language representation [52, 53] are suggested. Introducing domain descriptions and reference frameworks can help to unify used concepts and strategies for better development. A common understanding of terms may improve communication about the changes, accelerating development. Its proper usage may bring visualization into how the system works, simplifying the insight into the overall complex system.

Having a reference architecture may provide a new explanation of the system, somewhat mitigating the lack of causality in black box models. One participant also explicitly mentioned how reference architecture would help with improving traceability.

One of the reasons why reference architecture may not be possible is the sensitivity of the system. Companies do not want to make the details public as it would expose their approaches to the competition. This unwillingness was also shown during the workshop when the issue in the creation of reference architecture was not communicating all the details about the system. However, with open design or high-level reference architecture, the development of the system may be cheaper.

Additionally, it is difficult to describe concepts used in different companies. Each involved party may have a different understanding of terms and different approaches to describing the domain. Unifying all these definitions may require the creation of common language for the actual shared language.

Key findings of RQ3

- Shared language could help with better tailoring of requirements and collaboration.
- Shared language may make the process more standardized and provide a better explanation of the system.
- It might be difficult to create because of different concept understandings, the privacy of information, and competitiveness between companies.

6.4 Threats to validity

In this section the possible threats to validity of the thesis are discussed. We separate the threats to validity into three different types: internal validity, construct validity and external validity, and discuss them separately. The definition of the type are given in each subsequent section.

Construct validity

Construct validity refers to how good a study is at measuring and studying the concept or concepts it was designed to study. This subsection discusses the chosen methods and how they were designed to accurately measure and depict the intended concepts.

To design the interview guide the thesis followed the strategies and guidelines proposed by McNamara [71]. Additionally, the two supervisors supervising this thesis aided in the design of the interview guide. This was valuable for two main reasons. One, the supervisors had more knowledge regarding quantitative study design. And two, the supervisors had more knowledge regarding RE for the automotive industry. To help minimize the risk of asking biased questions and influencing the answers, the interview was designed to contain open questions. This allows the interviewees to answer freely and with reduced chance of the interviewers expected answers having an influence on the answers given. Furthermore, all interviewees were sent a short document previous to the interview, explaining the purpose and intent of the interview. Before the interview started, participants were asked if they had read the document, and if they needed further explanation regarding the subject matter and purpose of the interview. After a brief section regarding the interviewees experience and current role, the interview guide contained a terms & concepts section. During this section the interviewees were given the definition of relevant terms that would be used in questions throughout the interview. The interviewee was asked if they agreed with the definition, and if they did not, discussion would be had until a mutual understanding of the term was reached. This was done to avoid potential misunderstandings over differences in definition. Moreover, the interview guide includes a wrap-up question asking the interviewee if they feel that the interview has failed to include some subject or question that they feel is relevant to the topic, as can be seen in Appendix A.

Some of the interview participants were directly connected to the FAMER project. This could potentially lead to some bias in their confidence that the shared language of WP1 would help alleviate challenges. The inclusion of these FAMER participants was due to a difficulty in finding interviewees. Moreover, they are all automotive experts and it was deemed that their expertise was valuable enough to overcome the risk of a potential bias.

Moreover, the observed workshop was not designed or led by us. Therefore, there was a risk of misinterpretation. To mitigate this, we asked brief follow-up questions after the workshop.

Internal validity

Internal validity regards to whether there is a third factor affecting the results and influencing potential cause-and-effect relationships that have been identified.

During an interview study such as this one, there is a risk of researcher's bias. This means that preconceived notions and unstated hypotheses of the researchers may affect the data analysis and drawn conclusions. To mitigate this, continuous discussion was had with the two supervisors regarding the findings. Furthermore, both supervisors were part of the data analysis process, without being present during the interviews, to further minimize the risk of researcher's bias. Moreover, the thesis conducted a validation survey. Participants were asked to evaluate the identified themes, and whether they agreed with the results of the data analysis. We only managed to get three people to participate in the validation survey. It would have been preferable to have had more participants. However, due to time constraints, this was not possible.

External validity

External validity is about how generalisable the study's results are. As this is a case study, only concerning itself with the case companies, the results should not be expected to be applicable outside the case study context. The case study does however contain companies from multiple Tiers of the automotive supply chain. As such, it can be expected that the results are generalisable across the multiple Tiers of the specific case study companies.

7
Conclusion & future research

The thesis aimed to identify current approaches that work well and challenges regarding methods for requirement representation for automotive perception systems. The thesis also aimed to investigate whether a shared language could help alleviate potential challenges, without breaking any approaches that work well. To gather data, interviews were conducted with industry practitioners and experienced researchers. The data was then analysed by thematic analysis to identify key findings related to the three research questions. The findings were then verified by a validation survey.

The thesis explored the case study companies' current approach to requirement representation for automotive perception systems. Through this exploration, the approaches that are currently working well were identified. Moreover, the study identified challenges with representing requirements for automotive perception systems. Further analysis was done as to why certain approaches were considered challenging, many relating to the identified difficulties of RE for ML-enabled systems.

The thesis shows that there is a need for industry standards regarding ML-development for automotive perception systems, according to the case companies. The combination of defining the environment in which the system will be deployed with ODD's alongside defining specific requirements with natural language, has been identified as a well working approach. The thesis also identified that the case companies have a more formal approach towards communication in a multi-party context. This approach leads to trade-offs, carrying with it both positives and negatives. Moreover, the thesis has found that the case companies do not explicitly define fairness requirements. However, fairness-related factors are included implicitly in safety and performance requirements.

The shared language was unfortunately not completed in time to be fully tested in this thesis. However, by discussing the topic with interviewees, and observing a workshop regarding the creation of the shared language, some conclusions regarding its feasibility and potential could be made. The interviewees stated that a shared language could be helpful to alleviate challenges related to the lack of industry standards, and therefore help in defining a common development process. Some interviewees have raised concern regarding the feasibility of the reference architecture included in the shared language. They argue that the companies may not want to share sensitive system information during multi-party collaboration.

Once the shared language has been finalized in WP1 of FAMER, it would be interesting to further investigate its effect on the identified challenges and approaches that work well. The findings of this study could then be used to evaluate its effect on already defined challenges and approaches that work well. Moreover, in the pursuit of defining industry standards for ML-development in an automotive context, the challenges related to the lack of standards could be used as guidelines as to what issues the standards should address. Similarly, when the ISO standard *ISO/CD PAS 8800* is published, it would be interesting to conduct a similar study to see if standardization has in fact helped alleviate some of the challenges.

Bibliography

[1] G. Liebel, M. Tichy, E. Knauss, O. Ljungkrantz, and G. Stieglbauer, "Organisation and communication problems in automotive requirements engineering.," *Requirements Engineering*, vol. 23, no. 1, pp. 145 – 167, 2018.

[2] X. Zhao, K. Salako, L. Strigini, V. Robu, and D. Flynn, "Assessing safety-critical systems from operational testing: A study on autonomous vehicles.," *Information and Software Technology*, vol. 128, 2020.

[3] L. E. G. Martins and T. Gorschek, "Requirements engineering for safety-critical systems: A systematic literature review," *Information and software technology*, vol. 75, pp. 71–89, 2016.

[4] J. Dick, E. Hull, and K. Jackson, *Requirements Engineering.* Cham: Springer International Publishing, 2017.

[5] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach.* Pearson Education, 2 ed., 2003.

[6] E. Bjarnason, K. Wnuk, and B. Regnell, "Requirements are slipping through the gaps — a case study on causes & effects of communication gaps in large-scale software development," in *2011 IEEE 19th International Requirements Engineering Conference*, pp. 37–46, 2011.

[7] S. M. Ågren, E. Knauss, R. Heldal, P. Pelliccione, G. Malmqvist, and J. Bodén, "The impact of requirements on systems development speed: a multiple-case study in automotive.," *Requirements Engineering*, vol. 24, no. 3, pp. 315 – 340, 2019.

[8] H.-M. Heyn, K. M. Habibullah, E. Knauss, J. Horkoff, M. Borg, A. Knauss, and P. J. Li, "Automotive perception software development: An empirical investigation into data, annotation, and ecosystem challenges," in *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pp. 13–24, 2023.

[9] H.-M. Heyn, E. Knauss, A. P. Muhammad, O. Eriksson, J. Linder, P. Subbiah, S. K. Pradhan, and S. Tungal, "Requirement engineering challenges for ai-intense systems development.," *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), AI Engineering - Software Engineering for AI (WAIN), 2021 IEEE/ACM 1st Workshop on, WAIN*, pp. 89 – 96, 2021.

[10] D. Arruda and R. Laigner, "Requirements engineering practices and challenges in the context of big data software development projects: Early insights from a case study," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2012–2019, 2020.

[11] S. Marczak and D. Damian, "How interaction between roles shapes the communication structure in requirements-driven collaboration," in *2011 IEEE 19th International Requirements Engineering Conference*, pp. 47–56, 2011.

[12] K. M. Habibullah, H.-M. Heyn, G. Gay, J. Horkoff, E. Knauss, M. Borg, A. Knauss, H. Sivencrona, and J. Li, "Requirements engineering for automotive perception systems: An interview study," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 189–205, Springer, 2023.

[13] H.-M. Heyn, E. Knauss, I. Malleswaran, and S. Dinakaran, "An investigation of challenges encountered when specifying training data and runtime monitors for safety critical ml applications," in *Requirements Engineering: Foundation for Software Quality* (A. Ferrari and B. Penzenstadler, eds.), (Cham), pp. 206–222, Springer Nature Switzerland, 2023.

[14] A. P. S. Alves, M. Kalinowski, G. Giray, D. Mendez, N. Lavesson, K. Azevedo, H. Villamizar, T. Escovedo, H. Lopes, S. Biffl, J. Musil, M. Felderer, S. Wagner, T. Baldassarre, and T. Gorschek, "Status quo and problems of requirements engineering for machine learning: Results from an international survey.," 2023.

[15] R. Memon, K. Arezoo, K. Alipour, and M. Ghamari, "Autonomous driving systems: An overview of challenges in safety, reliability and privacy," in *2022 15th International Conference on Human System Interaction (HSI)*, pp. 1–7, 2022.

[16] Q. A. Ribeiro, M. Ribeiro, and J. Castro, "Requirements engineering for autonomous vehicles: a systematic literature review," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1299–1308, 2022.

[17] G. Ana, M. Kostadin, A. Ljupcho, and T. Dimitar, "Requirements engineering in machine learning projects.," *IEEE Access*, vol. 11, pp. 72186 – 72208, 2023.

[18] J. Hatcliff, A. Wassyng, T. Kelly, C. Comar, and P. Jones, "Certifiably safe software-dependent systems: challenges and directions," in *Future of Software Engineering Proceedings*, FOSE 2014, (New York, NY, USA), p. 182–200, Association for Computing Machinery, 2014.

[19] Y. V. Maksimov and S. A. Fricker, "Framework for analysis of multi-party collaboration," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pp. 44–53, 2019.

[20] N. Nahar, S. Zhou, G. Lewis, and C. Kästner, "Collaboration challenges in building ml-enabled systems: communication, documentation, engineering, and process," in *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, (New York, NY, USA), p. 413–425, Association for Computing Machinery, 2022.

[21] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, and F. G. de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *Journal of Systems and Software*, vol. 172, p. 110851, 2021.

[22] K. M. Habibullah and J. Horkoff, "Non-functional requirements for machine learning: Understanding current use and challenges in industry," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pp. 13–23, IEEE, 2021.

[23] H. Liu, "Trustworthy machine learning: Fairness and robustness," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1553–1554, 2022.

[24] T. ZEYU, Z. JIJI, and Z. KUN, "What-is and how-to for fairness in machine learning: A survey, reflection, and perspective.," *ACM Computing Surveys*, vol. 55, pp. 1 – 37, 2023.

[25] A. Balagopalan, H. Zhang, K. Hamidieh, T. Hartvigsen, F. Rudzicz, and M. Ghassemi, "The road to explainability is paved with bias: Measuring the fairness of explanations," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1194–1206, 2022.

[26] university of Gothenburg, "Facilitating multi-party engineering of requirements (famer)." `https://ffi-famer.github.io/`. Accessed: 2023-12-10.

[27] D. PESSACH and E. SHMUELI, "A review on fairness in machine learning.," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1 – 44, 2023.

[28] I. Sommerville, *Software engineering.* Pearson, 2016.

[29] "ISO 26262-1:2018, Road vehicles — Functional safety," standard, International Organization for Standardization, Geneva, CH, Dec. 2018.

[30] H.-M. Heyn, P. Subbiah, J. Linder, E. Knauss, and O. Eriksson, "Setting ai in context: a case study on defining the context and operational design domain for automated driving," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 199–215, Springer, 2022.

[31] P. Skruch, M. Szelest, M. Dlugosz, and D. Cieslar, "Safety of perception systems in vehicles of high-level motion automation," in *2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pp. 561–566, 2022.

[32] Q. Tang, J. Liang, and F. Zhu, "A comparative review on multi-modal sensors fusion based on deep learning," *Signal Processing*, vol. 213, p. 109165, 2023.

[33] S. Richard, *Computer Vision : Algorithms and Applications.*, vol. Second edition of *Texts in Computer Science.* Springer, 2022.

[34] L. Mendiboure, M. L. Benzagouta, D. Gruyer, T. Sylla, M. Adedjouma, and A. Hedhli, "Operational design domain for automated driving systems: Taxonomy definition and application," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–6, 2023.

[35] I. Häring and I. Häring, "Technical safety and reliability methods for resilience engineering," *Technical Safety, Reliability and Resilience: Methods and Processes*, pp. 9–26, 2021.

[36] M. Glinz, "On non-functional requirements," in *15th IEEE international requirements engineering conference (RE 2007)*, pp. 21–26, IEEE, 2007.

[37] T. Myklebust and T. Stålhane, *Functional safety and proof of compliance.* Springer, 2021.

[38] J. Vilela, J. Castro, L. E. G. Martins, and T. Gorschek, "Safety practices in requirements engineering: The uni-repm safety module," *IEEE Transactions on Software Engineering*, vol. 46, no. 3, pp. 222–250, 2018.

[39] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[40] "Artificial intelligence act." https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2021)698792.

[41] J. A. Crowder and C. W. Hoff, *Requirements Engineering: Laying a Firm Foundation.* Springer, 2022.

[42] J. M. Fernandes and R. J. Machado, *Requirements in Engineering Projects.* Cham: Springer International Publishing, 2016.

[43] R. S. Pressman, *Software engineering : a practitioner's approach.* McGraw Hill, 2005.

[44] S. Lauesen, *Software requirements: styles and techniques.* Addison-Wesley, 2002.

[45] D. M. S.Balaji, "Wateerfallvs v-model vs agile: A comparative study on sdlc," June 2012.

[46] I. Sarker, M. Faruque, U. Hossen, and A. Rahman, "A survey of software development process models in software engineering," *International Journal of Software Engineering and its Applications*, vol. 9, pp. 55–70, 11 2015.

[47] V. Rastogi, "Software development life cycle models-comparison , consequences," vol. 6, pp. 168–172, 2015.

[48] K. Petersen, C. Wohlin, and D. Baca, "The waterfall model in large-scale development," in *Product-Focused Software Process Improvement* (F. Bomarius, M. Oivo, P. Jaring, and P. Abrahamsson, eds.), pp. 386–400, Springer Berlin Heidelberg, 06 2009.

[49] M. Cohn, *User stories applied : for agile software development.* Addison-Wesley signature series, Addison-Wesley, 2004.

[50] A. Al-Rawas and S. Easterbrook, "Communication problems in requirements engineering: A field study.," United States: NASA Center for Aerospace Information (CASI), 1996., 1996.

[51] M. Raatikainen, T. Männistö, T. Tommila, and J. Valkonen, "Challenges of requirements engineering — a case study in nuclear energy domain," in *2011 IEEE 19th International Requirements Engineering Conference*, pp. 253–258, 2011.

[52] S. Japs, H. Anacker, L. Kaiser, J. H. SE-TRIP, R. Dumitrescu, and F. Kargl, "D-reqs: Determination of security & safety requirements in workshops based on the use of model-based systems engineering.," *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), Requirements Engineering Conference Workshops (REW), 2021 IEEE 29th International, REW*, pp. 412 – 414, 2021.

[53] M. Krammer, N. Marko, E. Armengaud, D. Geyer, and G. Griessnig, "Improving methods and processes for the development of safety-critical automotive embedded systems," in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, pp. 1–4, 2010.

[54] K. M. Habibullah, G. Gay, and J. Horkoff, "Non-functional requirements for machine learning: understanding current use and challenges among practitioners.," *Requirements Engineering*, vol. 28, no. 2, pp. 283 – 316, 2023.

[55] G. Giray, "A software engineering perspective on engineering machine learning systems: State of the art and challenges," *Journal of Systems and Software*, vol. 180, p. 111031, 2021.

[56] S. R. Kaminwar, J. Goschenhofer, J. Thomas, I. Thon, and B. Bischl, "Structured verification of machine learning models in industrial settings," *Big Data*, vol. 11, no. 3, pp. 181–198, 2023. PMID: 34978896.

[57] J. Henriksson, M. Borg, and C. Englund, "Automotive safety and machine learning: Initial results from a study on how to adapt the iso 26262 safety standard.," *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS), Software Engineering for AI in Autonomous Systems (SEFAIAS), 2018 IEEE/ACM 1st International Workshop on, SEFAIAS*, pp. 47 – 49, 2018.

[58] "ISO/CD PAS 8800, Road Vehicles - Safety and artificial intelligence," standard, International Organization for Standardization, Geneva, CH.

[59] A. Vogelsang and M. Borg, "Requirements engineering for machine learning: Perspectives from data scientists," in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pp. 245–251, 2019.

[60] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[61] S. Kochanthara, T. Singh, A. Forrai, and L. Cleophas, "Safety of perception systems for automated driving: A case study on apollo.," *ACM Transactions on Software Engineering & Methodology*, vol. 33, no. 3, pp. 1 – 28, 2024.

[62] N. Nahar, S. Zhou, G. Lewis, and C. Kästner, "Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process," in *Proceedings of the 44th international conference on software engineering*, pp. 413–425, 2022.

[63] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, 2016.

[64] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Ståhl, "The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson," in *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 348–356, IEEE, 2013.

[65] O. Salo and P. Abrahamsson, "Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum," *IET software*, vol. 2, no. 1, pp. 58–64, 2008.

[66] S. Dey and S.-W. Lee, "A multi-layered collaborative framework for evidence-driven data requirements engineering for machine learning-based safety-critical systems," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pp. 1404–1413, 2023.

[67] K.-J. Stol and B. Fitzgerald, "The abc of software engineering research," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 27, no. 3, pp. 1–51, 2018.

[68] R. K. Yin, *Case study research and applications*, vol. 6. Sage Thousand Oaks, CA, 2018.

[69] B. Flyvbjerg, "Five misunderstandings about case-study research," *Qualitative inquiry*, vol. 12, no. 2, pp. 219–245, 2006.

[70] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, pp. 131–164, 2009.

[71] C. McNamara, "General guidelines for conducting research interviews," *Retrieved May*, vol. 2, p. 2013, 2009.

[72] F. Shull, J. Singer, and D. I. Sjøberg, *Guide to advanced empirical software engineering*. Springer, 2007.

[73] J. Saldaña, *The Coding Manual for Qualitative Researchers*. Los Angeles, CA: SAGE Publications, 2nd ed., 2013.

[74] ISO, *ISO 15288:2023: Systems and software engineering — System life cycle processes*. Geneva: International Organization for Standardization, 2023.

[75] ISO, *ISO 21434:2021: Road vehicles - Cybersecurity engineering*. Geneva: International Organization for Standardization, 2021.

[76] "ISO/IEC 23053:2022, Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)," standard, International Organization for Standardization, Geneva, CH, June 2022.

[77] J. Birch, R. Rivett, I. Habli, B. Bradshaw, J. Botham, D. Higham, P. Jesty, H. Monkhouse, and R. Palin, "Safety cases and their role in iso 26262 functional safety assessment," vol. 8153, September 2013.

[78] P. Bishop and R. Bloomfield, "A methodology for safety case development," February 1998.

A
Interview guide

This appendix shows the interview guide that was used for conducting the interviews.

About us

Oskar Jakobsson & Zuzana Rohacova
Studying Software Engineering Master at Chalmers.
Currently doing our master thesis on "Requirement representation for safety-critical and fairness aware automotive perception systems"

Purpose of Study

The purpose of this study is to investigate challenges and current practices in requirements representation for safety-critical automotive perception systems. Specifically, for safety and fairness requirements. The goal of this interview is to get your personal experience and knowledge regarding the topic and to get insight into potential solutions.

Starting Questions

(10 minutes)

General Questions

Interviewee data

- What is your role in the company?
  - Describe shortly your role and what you work with?
  - How much relevant experience in years do you have with the development of perception systems in the automotive industry?
  - Are you working in management or in development?
  - Do you or did you actively develop software?
- Would you say that you are working mainly in research or in production / product development?
- What is your current relationship with requirements engineering in your company? For example, do you facilitate and define requirements or do you receive and apply them in development?

Company data

- In which markets in the world is the company active?
- Approximately, how many people work at your company?
- If your company works with suppliers, in which country/regions are they located?
- Where is your company situated in the automotive supply chain?
  - OEM
  - Tier 1 (supplies product directly to OEM)
  - Tier 2 (supplies Tier 1)
  - Others: Specify.

Terms and Concepts

- Define the following terms and concepts and ask if the interviewee agrees with the definition:
  - Non-functional requirements,
  - ML-component,
  - Safety-critical systems,
  - Fairness requirements,
  - Perception systems,
  - Related system or component (Sensors etc),
  - Reference architecture

Requirement representation

(10 minutes)

- In your work do you work with functional requirements for perception systems or related systems/components?
  - If they say no: How do they specify desired behavior?
  - If they say yes: Do they follow any standardized process for RE, such as those defined in ISO 26262 (for example SPICE software ..)?
  - If they say yes: How do you represent functional requirements for perception systems or related systems/components?
  - If they do not work directly with perception systems, what related system do they work with?
- What typical non-functional requirements do you get into contact with?
  - Can you rank them from the most to least important? (If not working with NFR then include slide with examples)
- How do you represent non-functional requirements for perception systems or related systems/components? For example use cases, ODD, natural language...
  - How do you represent the top 2 ranked requirements?

Multi-party collaboration

(10 minutes)

- How do you communicate functional and non-functional requirements for perception systems within your company?
  - What works well?
  - What does not work well?
- How do you communicate functional and non-functional requirements for perception systems with other companies in your supply chain?
  - Are you using the same methods as within your company or do you use methods from another company in your supply chain?
  - How do you figure which requirements representation to use between companies?
  - What works well?
  - What does not work well?

Machine learning specific challenges

(10 minutes)

- In a nutshell, how do you approach the integration of training data in your development process for perception systems?
- Do you see any challenges in representing requirements both functional and non-functional because of the introduction of ML components in the perception systems?
  - If additional challenges: What are or could be, in your opinion, potential solutions to these challenges?
  - Are there any additional non-functional requirements due to the introduction of ML-components?
    - If not working with fairness: Do you see any reason or benefit in defining fairness requirements for your systems?
  - Did the representation of both functional and nonfunctional requirements change because of the introduction of ML-components for your company?
  - How do you represent non-deterministic requirements for ML-components?

Potential solutions

(5 minutes)

- Do you think a common language between all companies in the supply chain could help alleviate the identified challenges of communicating requirements and requirements representation?
  - If yes, how? If not, why not?
- Do you think it is possible to have a clearly defined reference architecture for the perception systems across multiple companies in the automotive supply-chain?
- Do you think it would help working with and defining requirements across multiple companies in the automotive supply chain?

Wrap-up questions

(5 minutes)

- Is there anything important that you felt we missed? Any question or subject that would be relevant to the topic (of requirement representations in multi-party automotive system development)?
- Do you have any person in mind (from your organisation or outside of your organisation) that we should interview as well?

B

Identified categories and sub-themes from data coding

Table B.1 shows identifies sub-themes for each category.

Category	Sub-category	Sub-theme
Requirements representation:		
	Functional requirements representation:	FR experience
		FR challenges
		FR representation strategy
		FR comments
		Following standards
	Non-functional requirements representation:	NFR experience
		NFR challenges
		NFR representation strategy
		NFR practices
		NFR comments
Multi-party collaboration:		
	Incompany collaboration process:	Incompany process description
		Incompany challenges
		Incompany requirements representation
		What works well incompany
	Multi-party collaboration process:	Multiparty process description
		Multiparty challenges
		Multiparty requirements representation
		What works well multiparty
ML challenges:		
		Approach to integration training
		Representing ML requirements
		Additional challenges due to ML
		Additional requirements due to ML introduction
		Potential solutions for ML challenges
Fairness requirements:		
		How beneficial
		Relation to safety
Potential solutions:		
	Common language:	Advantages
		Challenges
		Comments
	Reference archietcture:	Advantages
		Challenges
		Comments

Table B.1: Identified sub-themes from data coding

C
Identified themes after coding

Figure C.1 and Figure C.2 show the first version of identified themes. Green boxes mark section, blue boxes mark theme and purple boxes marks comments.

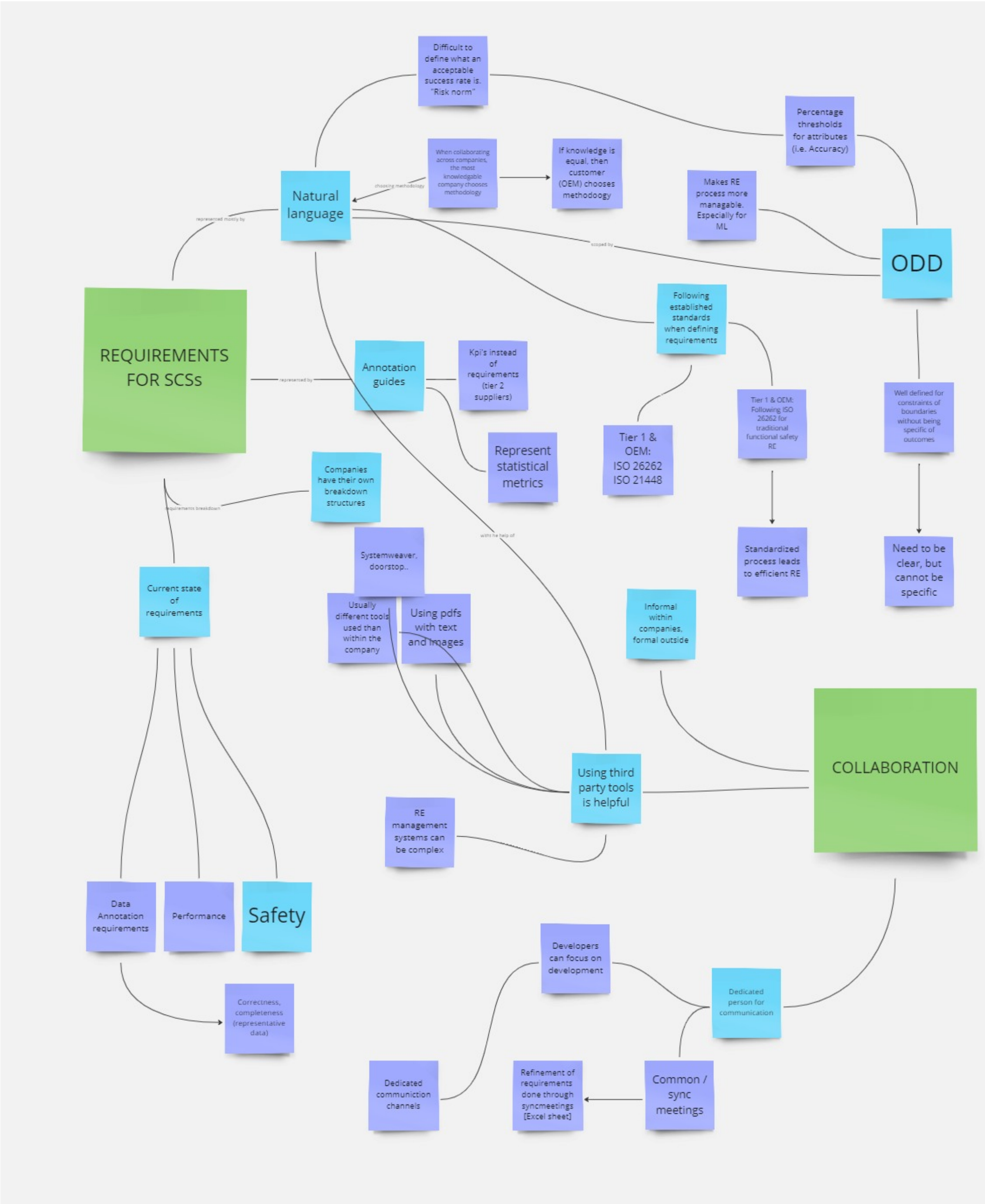

Figure C.1: First version of identified themes

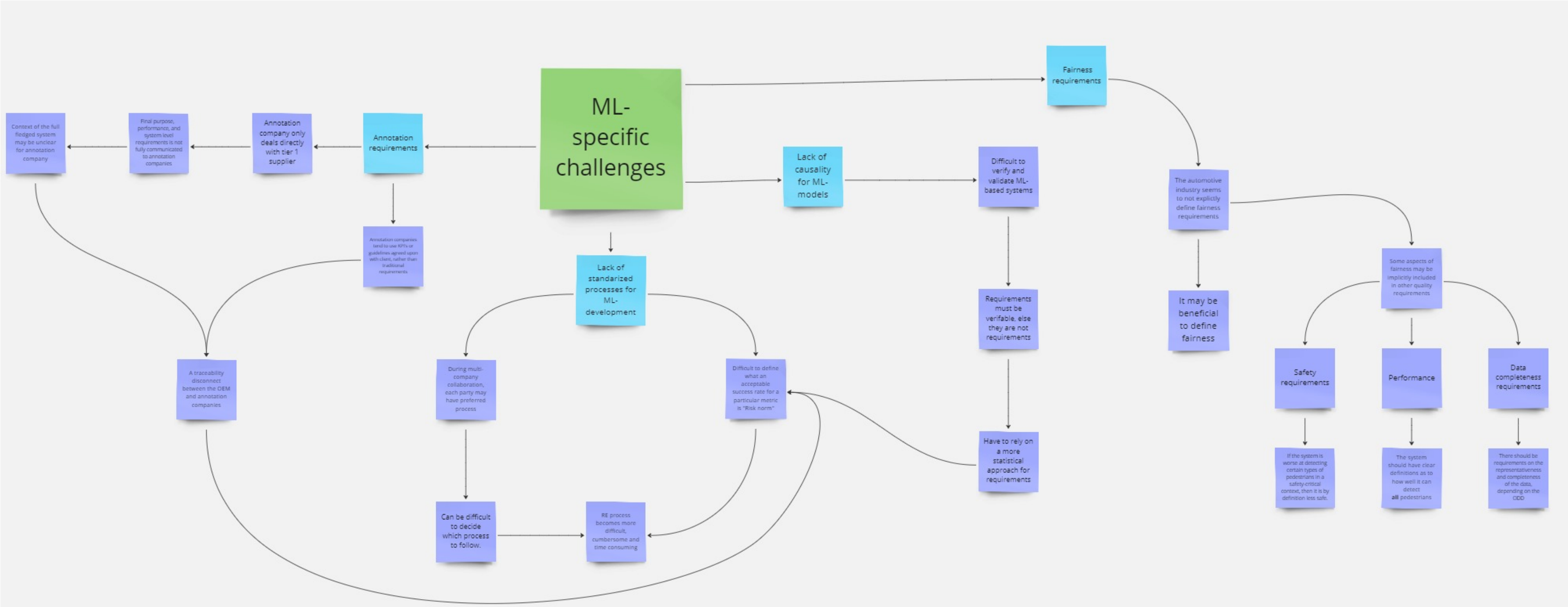

Figure C.2: First version of identified themes, cont'd

D
Challenges and approaches that work well

This Appendix contains a figure mapping out all the challenges and approaches that work well that were identified through thematic analysis of interviews, see Figure D.1.

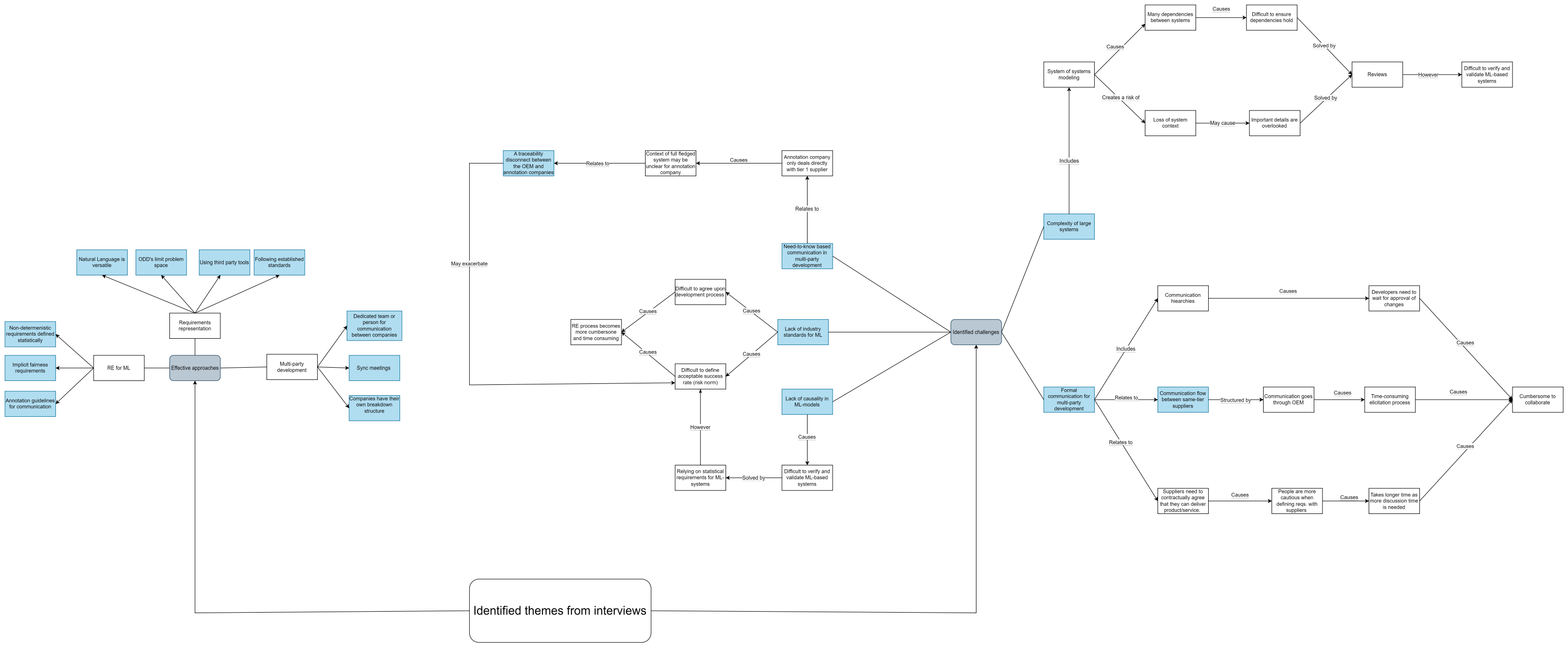

Figure D.1: Identified challenges and approaches that work well

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY