Thesis for The Degree of Doctor of Philosophy

An Empirical Investigation of Using Models During Requirements Engineering in the Automotive Industry

Grischa Liebel

Division of Software Engineering
Department of Computer Science & Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden, 2018

An Empirical Investigation of Using Models During Requirements Engineering in the Automotive Industry

GRISCHA LIEBEL

Copyright ©2018 Grischa Liebel
except where otherwise stated.
All rights reserved.

ISBN 978-91-7597-737-9
Doktorsavhandlingar vid Chalmers tekniska högskola, Ny serie nr 4418.
ISSN 0346-718X

Technical Report No 158D
Department of Computer Science & Engineering
Division of Software Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2018.

"Whoever has visions should go to the doctor."
- Helmut Schmidt

Abstract

Context: The automotive industry is undergoing a major transformation from a manufacturing industry towards an industry that relies heavily on software. As one of the main factors for project success, requirements engineering (RE) plays a major role in this transition. Similar to other areas of automotive engineering, the use of models during RE has been suggested to increase productivity and tackle increasing complexity by means of abstraction. Existing modelling frameworks often prescribe a variety of different, formal models for RE, trying to maximise the benefit obtained from model-based engineering (MBE). However, these frameworks are typically based on assumptions from anecdotal evidence and experience, without empirical data supporting these assumptions.

Objective: The overall aim of our research is to investigate the potential benefits and drawbacks of using model-based RE in an automotive environment based on empirical evidence. To do so, we present an investigation of the current industrial practice of MBE in the automotive industry, existing challenges in automotive RE, and potential use cases for model-based RE. Furthermore, we explore two use cases for model-based RE, namely the creation of behavioural requirements models for validation and verification purposes and the use of existing trace models to support communication.

Method: We address the aims of this thesis using three empirical strategies: case study, design science and survey. We collected quantitative and qualitative data using interviews as well as questionnaires.

Results: Our results show that using models during automotive RE can be beneficial, if restricted to certain aspects of RE. In particular, models supporting communication and stakeholder interaction are promising. We show that the use of abstract models of behavioural requirements are considered beneficial for system testing purposes, even though they abstract from the detailed functional requirements. Furthermore, we demonstrate that existing data can be understood as a model to uncover dependencies between stakeholders.

Conclusions: Our results question the feasibility to construct and maintain large amounts of formal models for RE. Instead, models during RE should be used for a few, important use cases. Additionally, MBE can be used as a means to understand existing problems in software engineering.

Keywords

Software Engineering, Empirical Research, Requirements Engineering, Modelling, Model-Based Engineering, Automotive Engineering

Acknowledgment

To a large extent, I owe the success of my 5-year PhD project to a number of people. These are:

My three supervisors, who all have played rather specific roles in getting me to graduate: Matthias Tichy, for hiring me and giving me the possibility to conduct research. Thanks for all the guidance and feedback, despite my tendency to ignore possible plans or expectations! Eric Knauss, my main supervisor, for taking over the baton from Matthias and "finishing the job" in a great and supportive way. Jörgen Hansson, for providing me with high-level views and strategic guidance, going beyond the meaning of individual papers. Finally, my examiner Aarne Ranta, for always being encouraging and very positive towards my decisions.

To all my colleagues at the Software Engineering Division, including of course all the administrative staff: thank you for providing a great work place!

My research would not have been possible without the support from all my collaborators in academia and industry. Therefore, I would like to thank the 46 individuals who co-authored papers with me. In particular, Gerald Stieglbauer and Oscar Ljungkrantz for welcoming my ideas and providing me with industrial resources and feedback.

I would like to thank Oscar Pastor, for first trying to recruit me as a PhD student (even though I already had a PhD position), but then still accepting that I "just" come over for a research visit.

Also, I would like to express my gratitude to Bran Selic, for always replying to my random questions about anything related to modelling and UML, motivating me to pursue my somewhat odd and non-technical studies on modelling, and for re-assuring me that one can and should avoid tool development during a PhD.

I am extremely thankful to have maintained a rather healthy work-life balance throughout the years. First and foremost, thank you Anna, for being at my side in all good and bad times, and for enduring my regular motivation and productivity swings - and for all the обнимашки.

Furthermore, I owe much of my success to my parents, who always encouraged me and let me make my own choices in life (e.g., to not study math).

Recently, *the band*, has provided a great counterpart to work. Thanks Carlo, Eugene, Pier, Enzo, Marco!

Finally, I would like to thank the Swedish weather. It is hard to describe how motivating it is to keep working when it is raining horizontally for days without end!

Funding: The research that led to the findings presented in Papers A, B, C, and D of this thesis received partial funding from the European Union's Seventh Framework Program (FP7/2007-2013) under grant agreement No 332830 and from Vinnova under DIARIENR 2012-04304.

List of Publications

Appended publications

This thesis is based on the following publications:

[A] G. Liebel, N. Marko, M. Tichy, A. Leitner, J. Hansson "Model-Based Engineering in the Embedded Systems Domain - An Industrial Survey on the State-of-Practice"
International Journal on Software and Systems Modeling 17(1), 91-113, 2018 (accepted 2016).

[B] G. Liebel, M. Tichy, E. Knauss, O. Ljungkrantz, G. Stieglbauer "Organisation and Communication Problems in Automotive Requirements Engineering"
Requirements Engineering 23(1), 145-167, 2018 (accepted 2016).

[C] G. Liebel, M. Tichy, E. Knauss "Use, Potential, and Showstoppers of Models in Automotive Requirements Engineering"
Under revision at International Journal on Software and Systems Modeling.

[D] G. Liebel, A. Anjorin, E. Knauss, F. Lorber, M. Tichy "Exploring Behavioural Requirements Modelling in the Embedded Industry"
In submission to a journal.

[E] M. Mohamad, G. Liebel, E. Knauss "LoCo CoCo: Automatically Constructing Coordination and Communication Networks from Model-Based Systems Engineering Data"
Information and Software Technology 92, 179-193, 2017.

Other publications

The following publications were published during my PhD studies. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not related to the thesis.

Journal (Peer reviewed)

[a] G. Liebel, H. Burden, R. Heldal "For Free: Continuity and Change by Team Teaching"
Teaching in Higher Education 22(1), 62-77, 2016

[b] J.-P. Steghöfer, H. Burden, R. Hebig, G. Calikli, R. Feldt, I. Hammouda, J. Horkoff, E. Knauss, G. Liebel "Involving External Stakeholders in Project Courses"
ACM Transactions on Computing Education (TOCE), 2018

Conference (Peer reviewed)

[a] G. Liebel, N. Marko, M. Tichy, A. Leitner and J. Hansson "Assessing the State-of-Practice of Model-Based Engineering in the Embedded Systems Domain"
17th International Conference on Model-Driven Engineering Languages and Systems (MODELS), 2014
Best Paper Award, Foundations Track

[b] G. Liebel, A. Olaru, S. Rajendran, U. Ingelsson, H. Kaijser, H. Lönn, R. B. Svensson "Addressing Complexity in Automotive Embedded Systems - Selection of System Model Elements for Allocation of Requirements"
4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2016

[c] G. Liebel, R. Heldal, J.-P. Steghöfer "Impact of the Use of Industrial Modelling Tools on Modelling Education"
29th IEEE Conference on Software Engineering Education and Training (CSEE&T), 2016

[d] M. Fagerström, E. E. Ismail, G. Liebel, R. Guliani, F. Larsson, K. Nordling, E. Knauss and P. Pelliccione "Verdict Machinery: On the need to automatically make sense of test results"
International Symposium on Software Testing and Analysis (ISSTA), 2016

[e] S. Karg, A. Raschke, M. Tichy, G. Liebel "Model-Driven Software Engineering in the openETCS Project: Project Results and Lessons Learned"
19th International Conference on Model-Driven Engineering Languages and Systems (MODELS), 2016

[f] G. Liebel, A. Anjorin, E. Knauss, F. Lorber, M. Tichy "Modelling Behavioural Requirements and Alignment with Verification in the Embedded

Industry"
5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2017

[g] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, B. Kanagwa "Requirements Engineering Challenges in Large-Scale Agile System Development"
25th International Requirements Engineering Conference (RE), 2017

[h] G. Liebel, O. Badreddin. R. Heldal "Model Based Software Engineering in Education: A Multi-Case Study on Perception of Tools and UML"
30th IEEE Conference on Software Engineering Education and Training (CSEE&T), 2017

Workshop (Peer reviewed)

[a] M. Tichy, C. Krause, G. Liebel "Detecting performance bad smells for Henshin model transformations"
2nd Workshop on the Analysis of Model Transformations (AMT), 2013

[b] C. Brenner, J. Greenyer, J. Holtmann, G. Liebel, M. Tichy, G. Stieglbauer "ScenarioTools Real-Time Play-Out for Test Sequence Validation in an Automotive Case Study"
13th International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT), 2014

[c] G. Liebel, M. Tichy "Comparing Comprehensibility of Modelling Languages for Specifying Behavioural Requirements"
First International Workshop on Human Factors in Modeling (HuFaMo), 2015

[d] E. Knauss, G. Liebel, K. Schneider, J. Horkoff and R. Kasauli "Quality Requirements in Agile as a Knowledge Management Problem: More than Just-in-Time"
2nd International Workshop on Just-In-Time Requirements Engineering (JIT), 2017

[e] F. G. de Oliveira Neto, J. Horkoff, E. Knauss, R. Kasauli, G. Liebel "Challenges of Aligning Requirements Engineering and System Testing in Large-Scale Agile: A Multiple Case Study"
4th International Workshop on Requirements Engineering and Testing (RET), 2017

[f] F. Bordeleau, G. Liebel, A. Raschke, G. Stieglbauer, M. Tichy "Challenges and Research Directions for Successfully Applying MDE Tools in Practice"
1st MDETools Workshop, 2017

Other (Not peer reviewed)

[a] G. Stieglbauer, M. Tichy, G. Liebel "Verfahren zur Durchführung eines Prüflaufs an einem Prüfstand mit einem Prüfling "
Patent IPC G01M 017/00, granted 2018

[b] N. Marko, G. Liebel, D. Sauter, A. Lodwich, M. Tichy, A. Leitner, J. Hansson "Model-Based Engineering for Embedded Systems in Practice" *In Research reports in software engineering and management 2014:01*

[c] G. Liebel, M. Tichy, N. Marko, J. Hansson "Industrielle Praxis modellbasierter Entwicklung im Bereich eingebetteter Systeme" *Multikonferenz Software Engineering & Management (SE), 2015*

[d] G. Liebel, R. Heldal, J.-P. Steghöfer, M.R.V. Chaudron "Ready for Prime Time - Yes, Industrial-Grade Modelling Tools can be Used in Education" *In Research reports in software engineering and management 2015:01*

[e] G. Liebel "Model-based requirements engineering in the automotive industry: Challenges and opportunities" *Licentiate Thesis, Technical Report No 146L, ISSN 1652-876X, Department of Computer Science & Engineering, Chalmers University of Technology and Göteborg University, 2016*

[f] S. Karg, A. Raschke, M. Tichy, G. Liebel "Model-Driven Software Engineering in the openETCS Project: Project Experiences and Lessons Learned" *Multikonferenz Software Engineering & Management (SE), 2017*

[g] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, B. Kanagwa "Requirements Engineering Challenges in Large-Scale Agile System Development" *Multikonferenz Software Engineering & Management (SE), 2018*

[h] M. Mohamad, G. Liebel, E. Knauss "LoCo CoCo: Automatically Constructing Coordination and Communication Networks from Model-Based Systems Engineering Data" *Multikonferenz Software Engineering & Management (SE), 2018*

Research Contribution

I joined the work on Paper A during the early planning phases of the survey. I consequently contributed by consolidating the existing survey and research questions and introducing the final study design. The hypotheses which are used for data analysis were elicited by me. Furthermore, I took the lead of executing the survey, analysing the data and writing the final publication.

My contributions to Papers B and C are the study design, data collection, data analysis and the majority of writing. The remaining co-authors contributed with reviews, interview organisation and improvement suggestions.

In Paper D, I did most of the work creating and improving the models. Additionally, I helped preparing and conducting the interviews and wrote the majority of the final publication. I was not actively involved in the test case generation.

Paper E builds on my ideas and is a direct result from the findings in Papers B and C. I established the contact to the case company and initiated the study. The study itself was then conducted by a Master student, the first author of the publication, under my continuous supervision. I wrote substantial parts of the final publication.

Contents

Chapter 1

Introduction

Since the first introduction of software in the automotive domain about 40 years ago, the amount of software in cars has grown exponentially [1]. Increasingly, costs and innovations in the automotive domain are driven by software [2]. While software is nowadays an essential element in automotive systems, it also causes interactions between parts of the vehicle that were traditionally independent [1], thus giving rise to the system's complexity.

In software engineering (SE), requirements engineering (RE) is known as one of the major factors affecting project success [3–5]. Specifically in the automotive domain, RE has been reported as a key challenge, e.g., in [1, 6]. Hence, targeting RE as an area of improvement could substantially improve the current SE practice in the automotive domain. Therefore, this PhD thesis targets the area of automotive RE as a problem domain.

The use of formal models in SE has long been advocated for, as a means to abstract from the problem and tackle complexity. The use of models throughout the SE lifecycle is known as model-based engineering (MBE) or model-driven engineering (MDE) [7]. In addition to handling complexity, there is evidence that MBE improves productivity, quality and other important factors [8, 9]. Therefore, using models also during automotive RE could be beneficial to tackle existing problems.

However, several challenges are associated with MBE, such as lack of tool maturity, tool interoperability, required education, and organisation support [9–12]. Using formal methods in the wider sense has also been debated heavily, in particular regarding adoption and usefulness in practice [13]. For example, Tony Hoare, arguably one of the most influential researchers in the area of formal methods, stated in 1995:

> *"Ten years ago, researchers into formal methods (and I was the most mistaken among them) predicted that the programming world would embrace with gratitude every assistance promised by formalisation to solve the problems of reliability that arise when programs get large and more safety-critical. Programs have now got very large and very critical - well beyond the scale which can be comfortably tackled by formal methods. There have been many problems and failures, but these have nearly always been attributable to inadequate analysis of requirements or inadequate management control. It has turned*

> *out that the world just does not suffer significantly from the kind of problem that our research was originally intended to solve."*
> - Sir Tony Hoare

Interestingly, this quote not only questions the use of formal methods for problems of industrial complexity, it also points to requirements as one of the main factors for failures. For RE specifically, Jackson [14] argues that RE is a "meeting place between formal and informal" and that RE cannot be purely formal. Hence, while issues in RE clearly seem to be worth addressing, the extent to which it is practical to adopt modelling techniques during RE is debated.

Regardless of this debate, existing solution proposals advocate a substantial use of formal models during RE, e.g., [6, 15]. However, this work is typically based on assumptions and anecdotal evidence, and there is only little empirical work investigating the use of MBE during RE.

Therefore, the overall aim of this PhD project is to investigate the potential use of model-based requirements engineering (MBRE) in an automotive context. To do so, the goals of this thesis are as follows.

G1: To empirically investigate the current and potential use of MBE during automotive RE.

G2: To explore solutions addressing existing RE challenges by using models.

Goal G1 aims to build a broad empirical basis on which future work can propose and evaluate MBE approaches to automotive RE. It includes understanding how MBE currently functions in the automotive domain, assessing which challenges exist in automotive RE, and for which areas of RE models could be most promising. Goal G2 constitutes the constructive part of this thesis as complement to G1, targeting a subset of the specific suggestions elicited for G1.

In vehicle projects, requirements are typically evolved and rarely start from scratch [16]. Therefore, in this thesis, we mainly consider the aspects of how requirements are specified and how existing requirements are managed. That is, elicitation of new requirements from stakeholders is not the main focus.

The remainder of this introduction chapter starts with an overview of the theoretical background of the problem domain, automotive RE, and of the solution domain, MBE, in Section 1.1. Based on the background, MBRE is discussed in Section 1.2 and the goals and scope of this thesis are re-visited in more detail in Section 1.3. Related work to this thesis is outlined in Section 1.4, followed by the research methodology in Section 1.5. The contribution of each individual paper is discussed in Section 1.6. The findings are summarised in Section 1.7, followed by a discussion of the overall thesis topic in Section 1.8. Threats to the validity of the thesis are discussed in Section 1.9. The introduction chapter is concluded by a summary and a discussion of future work.

1.1 Background

This thesis addresses the problem domain of RE as a part of automotive systems engineering. As a solution domain, MBE is considered. Therefore, this section provides an overview of automotive RE and MBE.

1.1.1 Automotive Requirements Engineering

According to Maurer and Winner [16], automotive systems engineering is “(A) methodology for developing systems for a vehicle, or a vehicle as a system”. This development includes several disciplines, such as mechanical engineering, electrical engineering and SE. Vehicles are usually developed by an original equipment manufacturer (OEM), in cooperation with several first-tier and second-tier suppliers.

Automotive systems engineering has specific characteristics, which distinguish it from other areas in systems engineering [16]. First, vehicles are used under greatly varying conditions, e.g., imposed by different laws in different countries, different skill levels and behaviour of drivers or variations within different cars of the same model. Secondly, demands on compatibility of subsystems are high, as components are re-used across vehicle models. In particular, this means that vehicle projects rarely start from scratch but rather evolve existing specifications. Thirdly, the high degree of safety critical functions and the large production volume greatly influence the costs of errors made during development. Finally, automotive systems engineering is traditionally organised in a vertical way [1]. That is, sub-systems are as independent as possible. The use of software, however, increasingly adds dependencies between these sub-systems [1].

Overall, the distribution among disciplines and organisations, as well as the specific characteristics of automotive systems engineering makes developing vehicles highly challenging.

The automotive industry faced rapid increases in size and complexity of the software included in vehicles in the past three decades [2]. For instance, in 2002, the Volvo XC90 automobile contained 38 Electronic Control Units (ECUs) [17]. The 2015 model of the same vehicle type already contains 108 ECUs [17], almost three times as many as in the 2002 model. Today, software is used for purposes such as engine control, infotainment and for safety-critical aspects of vehicles, such as braking or steering the vehicle. As a consequence, demands on the quality of software increase even more [2].

As a part of the overall development, the automotive industry is also struggling with RE [6]. From software projects on a general level, it is known that RE influences project outcome significantly [3]. Therefore, by improving RE, the chances of succeeding in a project can be increased. For instance, improving particular RE practices leads to fewer defects in the end product [18] or to increased productivity and improved communication during development [19].

RE is “a systematic and disciplined approach to the specification and management of requirements [..]” [20]. A requirement is “(1) A need perceived by a stakeholder. (2) A capability or property that a system shall have. (3) A documented representation of a need, capability or property.” [20]. The process of RE is typically broken down into requirements elicitation, requirements specification and requirements management. In requirements elicitation, requirements are sought, captured and consolidated [20]. This includes to address sufficiently the goals of multiple stakeholders, which can be conflicting. In requirements specification, the requirements are specified in a systematic fashion [20]. The resulting document is also called a requirements specification and separate documents can exist on the customer side (Customer Requirements Specification) and on the supplier side (System Requirements Specification) [20]. Finally, requirements management is the process of managing existing requirements specifications. In particular, this includes making changes to requirements and tracing later development artefacts, such as program code and tests, to requirements.

The overall requirements specification should meet several quality criteria to be considered a 'good' specification, according to IEEE Std. 830-1998 [21]. These contain (definitions according to [22]), e.g., correctness, to meet a customer's need or expectation; consistency, to not contain any conflicting requirements; or verifiability, that there exists a feasible way to check that the product meets the requirements. Some of these criteria are, in practice, difficult to achieve, e.g., completeness of a specification [22].

Overall, the specific characteristics and rapidly increasing complexity of software in automotive systems engineering, together with the importance of RE are the reason for selecting automotive RE as a problem domain in this thesis.

1.1.2 Model-Based Engineering

Models are central elements in many engineering disciplines. They can help to explain complex concepts in a simplified way, by excluding information that is not relevant for the explanation [7].

We use the following definition for a model, based on Stachowiak's features of a model [23]: *A model is a representation of entities and relationships in the real world with a certain correspondence for a certain purpose.* Therefore, when talking about a model, it is essential to answer what the model represents (the *model object*) and why it exists (the *model purpose*).

MBE and MDE are engineering approaches that have been devised to handle complexity and increase efficiency in the development of software or engineering in general [24]. As their names suggest, both employ models to handle complexity by means of abstraction. Additionally, both approaches are used in industry [10, 11, 25], and several empirical studies show benefits of MBE, e.g., increased productivity [11] and improved quality [26]. Therefore, we consider them as a candidate solution to address the challenges in automotive RE.

We use the definitions by Brambilla et al. [7], in which MBE comprises approaches were models play an important role but not necessarily the primary role. MDE has a narrower scope and is an approach in which models are used as the primary artefacts throughout the entire engineering process [7]. In the literature, many similar abbreviations are used to describe approaches that differ only slightly from MBE and MDE. For example, Model-Driven Development (MDD) can be seen as a subset of MDE, only focusing on software or systems development. Finally, Model-Driven Architecture (MDA) is a specific version of MDD devised by the Object Management Group [27], and therefore a subset of MDD. The overlap between the different approaches is visualised in Figure 1.1.

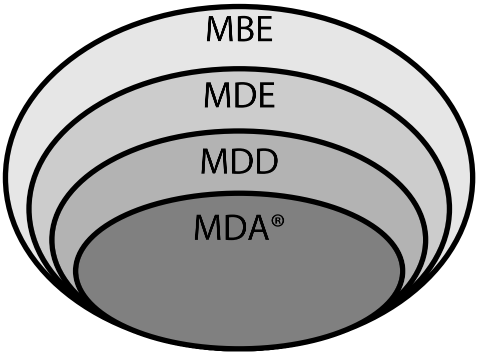

Figure 1.1: Classification of Model-Driven and Model-Based Approaches, Adapted from [7]

In model-centric approaches, the concept of a meta model plays a key role. Each model conforms to a meta model, it is "written in the language of each meta model" [28]. Creating new meta models allows essentially the creation of new (modelling) languages, a concept heavily utilised for so-called Domain-Specific Languages (DSLs). One of the most well-known, standard meta models is the Unified Modeling Language (UML) [29].

Apart from models and meta models, transformations play an important role in model-centric approaches. For example, model transformations are considered to be "a key part of MDA" [30] and to be "among the most important operations applied to models" [31]. Their task is to transform models into different artefacts, e.g., into other models or into program code. These transformation steps could be automated by using special transformation languages, e.g., following the Query/View/Transformation standard [32].

Ideally, transformations are automated in order to require as little as possible manual work in between the transformation steps. If this is the case, tracing between the models exists and complexity is reduced, as less manual work is needed to create and maintain artefacts throughout the entire engineering process. The overall vision of MDA, and typically of MDE, is then that models are created at all steps in the engineering process and transformed into other models, down to the final artefact, typically program code.

As to what extent model transformations are used in industry and whether it is in practice possible and feasible to have an automated chain all the way to program code, or any other low-level artefact, is not answered by related academic work. This vision is however not very realistic given the abstraction gaps in between artefacts in the development process, in particular between abstract artefacts that contain uncertainty and the final system. For example, high-level requirements do typically not contain any information on how the architecture of a software system should be structured. Therefore, domain knowledge is needed to add this information while constructing a model of the software architecture. An automated transformation would not be able to add this domain knowledge.

The terminology regarding model-centric approaches is not used consistently in academic literature. For example, the definition itself does not clearly state what it means that models are *primary* artefacts. Furthermore, some researchers regard only approaches which contain model transformations to be model-driven. Therefore, we position ourselves broadly in this thesis and use the most permissive term, MBE, as a possible solution domain to challenges in automotive RE. Our primary intention is not so much to outline exactly which of the different model-centric approaches should be used in practice, but rather to show different possibilities.

1.2 Model-Based Requirements Engineering

While the overall demands for developing automotive systems increase, requirements in particular are a major cost driver in all types of embedded systems [2]. Therefore, special attention towards the process of RE is warranted.

The documented benefits of using MBE in industry, e.g. in [10, 11, 26], indicate that MBE can be used to address these increased demands in industry. Given the importance of RE, MBE should not be restricted to software design and development only, but be already used during RE in the automotive domain. In this thesis, we refer to this approach as automotive MBRE. Figure 1.2 visualises the relationship between the problem domain of automotive RE as a subdomain of automotive systems engineering, and the solution domain of MBE.

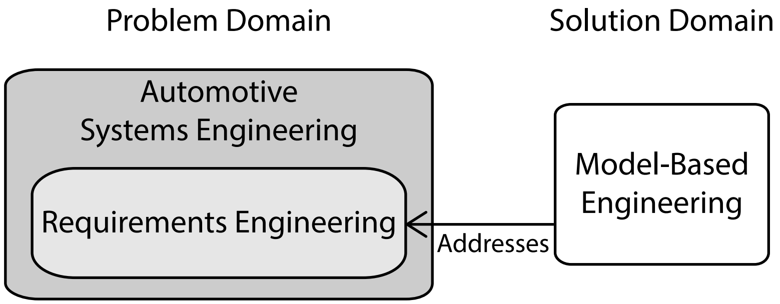

Figure 1.2: Model-Based Requirements Engineering

Based on the definitions of RE and MBE and the different aspects of automotive systems engineering, the 'Addresses' arrow in Figure 1.2 could be refined in several different ways. In particular, several approaches of using models in RE have been proposed in the past and, therefore, some kind of classification scheme or taxonomy is needed to clarify the nature of our contributions. For this purpose, we use the classification model depicted in Figure 1.3, using the feature model notation by Kang et al. [33]. In this figure, a feature can be seen as one aspect of model usage. We call one instance of this model a *configuration*. Some of the model's aspects can only have pre-defined values, i.e., *grade of formality* and *completeness*. Other aspects can have arbitrary values, or at least a non-exhaustive list of them, e.g., the *purpose*.

This classification is based on our current understanding. As of now, it has not been validated and serves therefore only as a means to describe the variety of how MBE can be used to address automotive RE (or other problem domains) and the contribution of this thesis. Therefore, we do not discuss here if all aspects are indeed required (mandatory) and whether or not the listed aspects exhaustively describe MBRE.

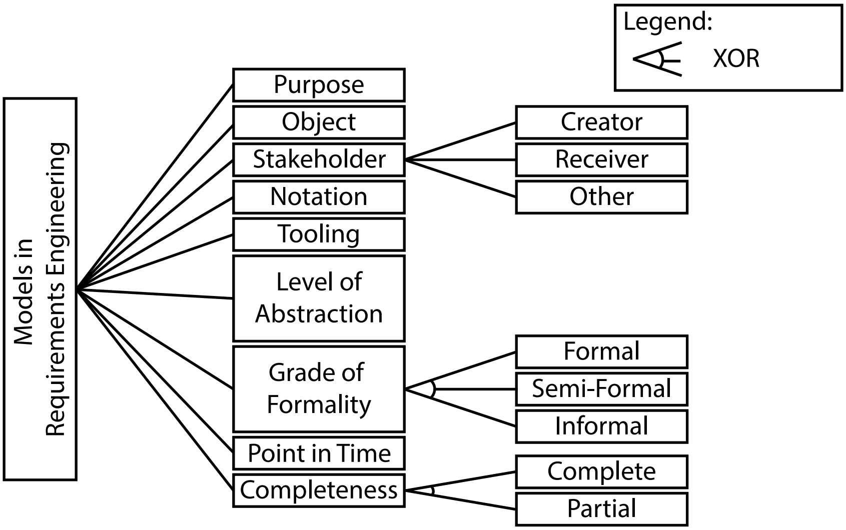

Figure 1.3: Different Aspects of using MBE for a Given Problem Domain

According to the definition, every model has a *purpose*. In MBRE, the model purpose could be, e.g., to increase the domain understanding, to document requirements in form of a graphical model, to aid elicitation, or to enable requirements validation.

Furthermore, each model represents a part of the real-world, which we here call the model *object*. For example, a model could represent a functional requirement, an entire requirements specification, the relation between multiple stakeholders used for elicitation, or the structure of the requirements specification and traces between single requirements. Whether functional requirements or non-functional requirements (quality requirements) are described is also dictated by the model *object*. Due to the rather important distinction of these two terms, they could, however, also be seen as a separate aspect in this classification.

Similarly to requirements and requirements specifications, each model in MBRE has a number of *stakeholders*. These can be broken down further into the *creators* of the model, the *receivers* of the model and any *other* stakeholders. Creators need to have the technical and domain knowledge to create the models in a way that is semantically and syntactically correct and represents the domain in a correct way. Receivers need to be able to understand the model. In RE, both receivers and creators are primarily domain experts, which means that they might not be experts in modelling. Especially when models are intended for documentation and comprehension purposes, this needs to be taken into account when devising modelling approaches for RE. Further stakeholders (*others*) can be considered in the RE sense, i.e., any person who has a stake in the model. This includes, e.g., internal stakeholders in the organisation, customers, end-users, as well as standardisation and regulation bodies.

Models can have different *notations*, which, in turn, can be graphical or textual. Candidates are common modelling languages, such as UML [29] or i* [34]. The notation could further be divided into standard and non-standard notations. However, this aspect is not relevant for the contents of this thesis and therefore omitted.

The *tooling* used for creating, editing and viewing models plays an important, and in practice often deciding, role. Tools influence which notations can be used and for what purposes the models can be employed. Furthermore, they heavily influence how the model can be accessed and distributed, e.g., through interchange with other tools or export functions.

The *level of abstraction* on which the models are created is especially relevant in RE. For example, a model of requirements can follow common requirements abstraction levels, such as the Requirements Abstraction Model [35] or Lauesen's classification into goal-level, domain-level, product-level and design-level requirements [22]. Additionally, a model could be intentionally simplified, e.g., to be more understandable to non-experts.

The *grade of formality* varies largely between different models. A model that is aimed at exploring the domain during requirements elicitation and that should be understandable for several types of (non-expert) stakeholders will most likely be informal, whereas a model used for simulation or code generation needs to be formal in order to fulfill its purpose. A model's grade of formality is typically broken down into *formal*, for defined formal syntax and semantics, *semi-formal*, when either semantics or syntax are not completely formally defined, or *informal*, when both are at least partially missing.

During RE, there is a high level of uncertainty initially, with only few requirements and goals being known. During requirements elicitation, this situation changes as uncertainty is slowly decreasing. Similarly, the amount of requirements, their abstraction level and the level of detail changes. Therefore, the *point in time* at which models are created plays an important role in MBRE. For example, goal models might be commonly created in early stages of RE, whereas detailed UML models for simulation are more realistic later on. Specifically to automotive systems engineering, requirements are rarely written newly, but rather evolve from project to project [16].

The term *completeness* can be defined in two different ways in MBRE. On the one hand, completeness is used in RE as a quality criterion to describe whether a requirements specification covers all non-trivial stakeholder expectations and needs. On the other hand, the notion of completeness is often used for models that contain all necessary information and do not conflict with constraints imposed by the meta model. Incompleteness in models can be caused by uncertainty or multiple stakeholder opinions [36].

We do not see uncertainty as a separate aspect in this classification. While it could be considered as such, we consider it a part of the model *notation*, i.e., notations that contain elements to explicitly encode uncertainty, and part of the *completeness*, i.e., by leaving out uncertain information.

The aspects depicted in Figure 1.3 are related to each other and influence each other in various ways. For instance, the most obvious relationship is that of *notation* to *tooling*, as a modelling tool supports only a defined set of modelling notations. Similarly, *notation* and *tooling* will in practice affect the *purpose*, as both can have different restrictions on what the resulting models might be used for. For example, it would make little sense to model behaviour requirements for the purpose of simulation using notation or a tool that does not support simulation/execution.

Additionally, the order in which the aspects are defined is often unclear. For example, organisations might decide to introduce a special model *notation* for a specific *purpose*, e.g., executable state machines in order to simulate system behaviour early on. However, this decision could be taken the other way around, e.g., if executable state machines are already used in the organisation, the decision to use them for simulation could be taken later. These relations between aspects are non-trivial and require further empirical investigation as a part of future work.

1.3 Goals and Scope

The classification of MBRE in Section 1.2 now allows us to revisit the goals of this PhD thesis and illustrate the contribution in terms of this classification.

As stated in the beginning of this chapter, the goal of the overall PhD thesis is to investigate the potential use of MBRE in an embedded systems context, mainly focusing on the automotive domain. In order to point out important factors in MBRE, this includes to a large extent building an empirical basis of the state of practice. In terms of Figure 1.3, this means that the aim is to investigate which configurations of the feature model already exist in practice (and possibly outside of RE), which ones are considered useful/promising by practitioners, and which ones could help address existing challenges in RE. In addition to understanding the current state, it is desirable to propose concrete solution candidates, i.e., explore configurations of the feature model.

While, ideally, it would be desirable to cover as much as possible of the current state in the automotive domain, this is unrealistic. Therefore, most of the studies addressing the aim of this thesis aim to investigate specific situations at our case companies. In particular, this is intended to provide an in-depth view on the current state, instead of trying to be general. This is further discussed in Section 1.5.

In terms of goals and research questions, the contribution of this PhD thesis is as follows.

G1: To empirically investigate the current and potential use of MBE during automotive RE.

> **RQ1:** To what extent is MBE used in the automotive domain and how is it perceived?
>
> **RQ2:** Which problems exist in automotive RE?
>
> **RQ3:** What is the current state and the potential of using models during automotive RE?

G2: To explore solutions addressing existing RE challenges by using models.

> **RQ4:** To what extent can real-life textual specifications of behaviour be translated to formal models?

RQ5: How can existing requirements specifications be exploited for improving communication (regarding requirements)?

G1 aims to form an empirical basis of MBRE in the automotive domain. More detailed, **RQ1** aims at establishing a picture of the different configurations of the model in Figure 1.3 which are already used in the automotive domain, but not restricted to RE only. **RQ2** is only indirectly related to the feature model in Figure 1.3. That is, it outlines problematic areas of RE, which could then be addressed by specific modelling solutions (e.g., by addressing specific *purposes* or model *objects*). **RQ3** aims at increasing the understanding of which kind of configurations of the model in Figure 1.3 could be beneficial for automotive MBRE.

The second goal, **G2**, explores specific configurations of the model in Figure 1.3, based on the findings for the first part of the thesis. This can be seen as a constructive complement to the empirical basis this thesis builds.

The goals are addressed by the five papers included in this PhD thesis, each addressing one of the five research questions. Figure 1.4 depicts how the papers address each goal and research question individually. Papers A, B, and C jointly address goal **G1**, forming an empirical basis of MBRE in the automotive domain. On top of this empirical basis, two concrete solution candidates are investigated, addressing goal **G2**. These are (a) the use of modelling languages for expressing behaviour requirements as models (Paper D, **RQ4**) and (b) the use of existing models of requirements specification to tackle existing RE challenges (Paper E, **RQ5**). Direction (a) is concerned with the creation of requirements models and resulting analysis possibilities, whereas direction (b) is more concerned with how existing requirements specifications can be understood as a model and what potential benefits result from this understanding. These two directions are investigated to show the potential benefit of MBRE based on two concrete cases, strongly supported by the empirical basis.

While the five research questions are primarily answered by one paper each, there is some overlap. In particular, Papers A to C give partial answers to all **RQ1**, **RQ2**, and **RQ3**.

1.4 Related Work

Given the broad scope of this PhD thesis, there is a substantial body of knowledge related to each of the research questions individually. Therefore, the detailed related work is discussed in each paper separately. Here, we give an overview of related work, both targeting individual goals and the overall thesis scope.

RQ1 targets the use of MBE in the embedded industry. Apart from our own contribution, several empirical studies exist that focus on the use of models in industry, e.g., [8–12, 25, 37]. Common themes in these studies are that MBE helps finding defects early [8, 38], increases productivity [8, 26], and improves the overall quality of the resulting system [25, 26].

However, a substantial amount of challenges is reported as well, the most common ones being tool related issues, e.g., lack of tool maturity [8, 9, 12, 25, 37]

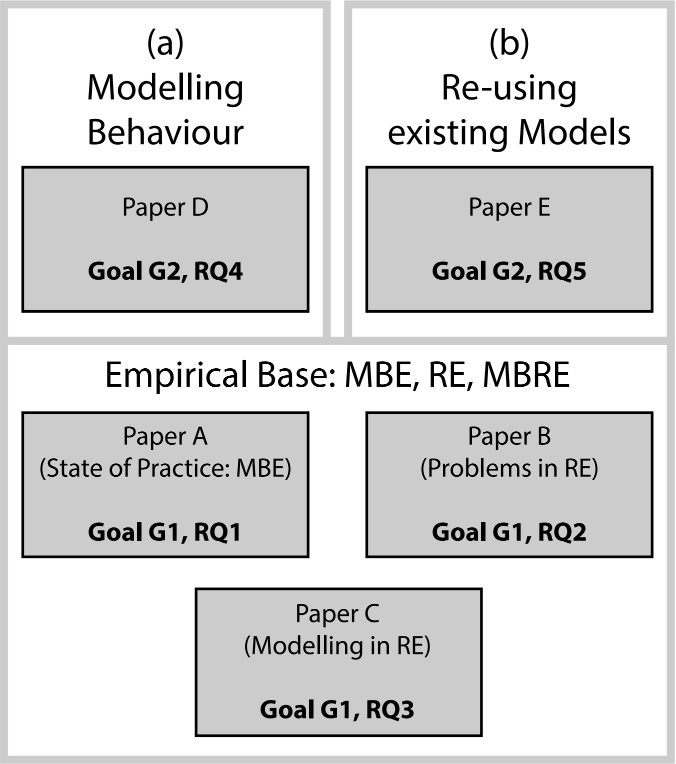

Figure 1.4: PhD Scope

and interoperability [8, 25, 39], and the complexity and lack of scalability of creating models [8, 9, 25, 39].

To succeed with MBE, existing work suggests that an iterative approach is needed [10], together with organisational committment [10, 12, 37, 40]. However, there is a need for substantial education of engineers [11].

Finally, there are few studies that go beyond the population of modelling proponents and investigate the extent to which modelling is used by software engineers in general, e.g., [41, 42]. They report a low use of models, in particular formal models. Reasons for not using models are that they are considered too much effort [41, 42] given the benefit, lack good tool support [42], or lack of usefulness [41].

Studies focusing explicitly on embedded systems, which is typically seen as a domain with large adoption of model-based techniques, are much less common than general studies on MBE adoption. Apart from our own studies, we are only aware of three studies focusing explicitly on MBE in embedded systems, i.e., [26, 38, 43]. Agner et al. [26] survey the Brazilian embedded industry, reporting that MBE increases productivity and improves quality, maintenance and portability. Additionally, the authors report that MBE is mainly used for documentation, with only little use of code generation or model-centric approaches. From a case study within the automotive domain, Kirstan and Zimmermann report positive effects of MBE, such as an earlier detection of errors or cost savings during initial development phases [38]. As shortcomings, they report tool interoperability. Finally, Heldal et al. [43] investigate the use of prescriptive models (i.e., models of something that does not yet exist) and descriptive models (i.e., models of something that does exist) at three embedded case companies. The authors report several model uses for different purposes, e.g., sketches of requirements to aid understanding of requirements. Additionally, the authors find that multiple languages and tools are necessary and, therefore, formulate the hypothesis that tool interoperability is a key challenge.

RQ2 targets problems in automotive RE. Several publications state that RE is one of the largest problems in this domain [1,6], and that RE is currently performed in an ad-hoc fashion [6]. While some of the authors have substantial experience and knowledge of automotive systems, their statements lack empirical support. We are aware of two empirical studies reporting specific problems in automotive RE [44, 45]. Almefelt et al. [45] report that requirements are often incomplete or conflicting, and that it is difficult to overview specifications due to their size. Pernstål et al. [44] report that requirements are often unclear in early phases of a project and that it is difficult to communicate requirements to suppliers.

Outside the automotive domain, several publications explore challenges in RE with respect to organisation and communication, e.g. in large-scale RE [46] and market-driven RE [47]. However, their findings are not domain specific. Thus, while it is unclear to what extent they apply to automotive RE, they serve as a frame of reference for **RQ2**.

RQ3 is specifically investigating the combination of models during RE. Models play an important role in the RE community, as can be seen from the many papers discussing models at the RE conference series [48], the premier academic conference in RE. As the classification in Section 1.2 suggests, the amount of possible combinations for applying modelling during RE is vast. Hence, it is not surprising that this is reflected in the amount of work in the RE community. Popular approaches are, in particular, goal modelling and meta modelling, i.e., creating new domain-specific languages [48].

Restricting the scope to automotive RE only, models are often seen as a hope to cope with future challenges, e.g., in [1, 49]. A common approach is to use structural modelling notations, such as EAST-ADL [50] or the SysML requirements diagram extensions [51], to enforce the structure of a requirements specification. For example, Boulanger and Văn [52] describe a methodology to develop embedded automotive systems, using EAST-ADL and SysML for requirements modelling. Similarly, Piques and Andrianarison [53] report industrial experiences with using SysML in the automotive domain. Albinet et al. [54] introduce a similar approach, but also use the UML profile MARTE for real-time systems. All three approaches have in common that the requirements themselves are expressed in natural language, whereas the model *object* is the specification structure.

With respect to industrial practice, the extent to which models are used during RE is not as clear. Lubars et al. [55] report that Entity-Relationship diagrams and object-oriented models are common in RE during the early 90s. However, the authors do not report how these models are used later on. The popularity of models could partially stem from the widespread use of the

Rational Unified Process and Structured Analysis at that time, which both include the use of models during RE. However, a more recent study by Sikora et al. [56] reports that practitioners in the embedded industry advocate a more intensive use of models during RE. The authors attribute this to the automation possibilities that RE models could offer. Finally, Méndez Fernández et al. [57] study how RE is executed in 12 successful[1] real-life projects at CapGemini. The authors analyse which artefacts are produced throughout the RE process and why. Results show that domain and environment models are present in all 12 projects.

For the remaining research questions, **RQ4** and **RQ5**, there is a substantial body of knowledge proposing solutions on how models can be used in industry, both in the automotive industry and beyond. The SPES 2020 Methodology [15] for the development of embedded systems and REMsES [6] are two prominent examples that prescribe the use of a large variety of different models throughout the entire development cycle. For instance, SPES 2020 suggests using context models, goal models, and scenario models during RE. Substantial follow-up work explores the use of SPES or adapts it for different types of systems, e.g., [58–60]. Interestingly, many of the findings in this follow-up work question the feasibility of introducing the methods on a large scale. For example, Böhm et al. [59] state that one of the main success factors of modelling a train control system was the existence of a high-quality, textual input specification. Similarly, Brings et al. [60] find that a large number of dependencies cause problems when trying to model cyber-physical systems.

Apart from SPES and REMsES, there are several other proposals for requirements modelling. For instance, Brandstetter et al. [61] propose early validation of requirements by means of simulation. Fockel and Holtmann [62] propose the use of controlled natural language and formal models in an interchangeable manner using bi-directional graph transformations. However, their solution proposal lacks empirical evaluation.

Focusing on verification only, models have been shown to enhance the testing process in what has been coined Model-Based Testing [63]. However, it is also common in this area of research to focus on solution proposals, while evaluation or validation research is scarce [63].

Overall, the related work on this PhD thesis can be summarised as follows. In recent years, a substantial body of knowledge regarding the use of MBE in industry has developed. Among the published studies, those addressing the embedded systems domain in particular are however scarce. Similarly, the use of models during RE specifically is rarely investigated empirically, even though solution proposals are common. Therefore, there is a need for a thorough investigation of the current use, the obstacles, and the potential of MBE during RE, considering the context of the automotive domain. We address this need in Papers A, B, and C.

While there is already substantial work proposing the use of models during RE, our work on goal G2 complements these by exploring aspects that have so far not been considered in detail. That is, exploring the use of high-level abstractions of requirements (as investigated in Paper D), and considering existing data as a structural model (as exploited in Paper E).

[1] released and deployed systems, used in production

1.5 Research Methodology

SE involves human activities as part of the development process. In particular, the development of software is a creative process and we are unable to 'manufacture' software [64]. Therefore, it is difficult to evaluate many aspects of SE without human involvement. For example, introducing a method that almost exclusively focuses on the use of models might work very well in theory, and result in substantial quality improvements, but engineers might still resist this introduction. This would render this good theoretical solution infeasible in practice. To acknowledge this, the research methodology used in this thesis follows the scientific method discussed by Basili [65]. In this method, the existing state is observed, followed by a proposal of a model or a theory, which is then evaluated [65]. In particular, the empirical paradigm uses empirical studies to do so [65]. Papers A, B, and C in this thesis aim to collect empirical data in order to understand the current situation in industry. This corresponds to the first point in the scientific method, "observe the world" [65]. In Papers D and E, methods to solve existing problems found in the first three papers are then proposed and evaluated.

We use different research strategies for the five studies described in the attached papers. An overview of the studies with their respective strategy and data collection methods is listed in Table 1.1.

Table 1.1: Research Strategies per Paper

Paper	Strategy	Data
Paper A	Survey	113 survey participants
Paper B	Case Study & Survey	14 interviewees, 31 survey participants
Paper C	Case Study	14 interviewees
Paper D	Design Science	3 interviewees, analytical evaluation
Paper E	Design Science	15 interviewees, 12 survey participants

In Paper A, we collected data using a survey. Surveys can be helpful to obtain a snapshot of the current situation and can give a broad overview of the surveyed area [64]. They are used to obtain a representative picture of a larger population [66]. This fits well the scope of a first, broad study, as we obtain a picture of MBE as a whole within the embedded systems domain. We do not yet focus on a specific problem (and not on RE specifically), but rather try to understand what is currently happening in the automotive domain.

Papers B and C are aimed at a more detailed situation, i.e., the requirements process in the automotive domain and the use of models within this process. To investigate this situation, we use a multiple case study, collecting qualitative data in the form of semi-structured interviews[2]. A case study is appropriate when the boundaries between the studied concept and the context are not clear [64], which is the case in Papers B and C. In particular, it is important to point out that we do not aim to generalise as broadly as we do in Paper A.

[2] Note that Papers B and C use the same interview data.

Instead, we try to understand the detailed situation in the context of study. In addition to the interview data, Paper B makes use of a follow-up survey aimed at testing the outcomes of the qualitative data analysis using a broader sample of practitioners.

Papers D and E make use of the design science research strategy. In design science research, new artefacts are created and evaluated in an iterative fashion [67]. That is, the main activities consist of designing the artefact and investigating it in its context [68]. In contrast to case studies, design science focuses on the creation of a new artefact instead of understanding a situation in its context (without actively changing it). In contrast to the action research strategy [69], design science focuses more strongly on the actual artefact, not necessarily changing the process or the context the artefact exists within.

In addition to choosing the research strategies to reach the intended aim of each study and the overall PhD thesis, there is an underlying thought of demonstrating literacy in research methodology towards the PhD degree. Therefore, we make use of a variety of different research strategies, covering case study (Papers B and C), survey (Papers A and B) and design science research (Papers D and E). Furthermore, the different papers contain elements of qualitative (Papers B, C, D, and E) as well as quantitative research (Papers A, B, and E).

Focusing on the author of this PhD thesis, there is also a different level of seniority throughout the papers, ranging from a participant role, mainly designing and conducting a study based on others' ideas (Paper A), over studies that were planned, designed and conducted to a large extent individually by the PhD candidate (Papers B, C, and D), to a study that was supervised by the PhD candidate, based on his own previous research (Paper E). Hence, the PhD thesis does not only contain a research contribution in itself, but demonstrates the succession towards the ability to perform independent research.

A further, more detailed discussion of the research strategies for the individual papers is found in Chapters 2 to 6.

1.6 Contribution

In the following, we outline the five papers included in this thesis shortly. The entire papers can be found in Chapters 2 to 6.

1.6.1 Paper A, State-of-Practice of MBE in Embedded Systems

MBE aims at increasing the effectiveness of engineering and handling complexity by using models as important artefacts throughout the entire engineering process. A substantial amount of empirical studies investigates the application, benefits and drawbacks of MBE in industry. However, there is a lack of comparable studies focusing on the area of embedded systems.

Modelling standards such as MARTE [70] and the widespread use of modelling tools such as Matlab/Simulink suggest that MBE is used widely in this domain. Personal experience from cooperation with industry corroborates this view. Nevertheless, empirical data is missing.

The contribution of Paper A is to fill this gap by providing empirical data on the state of practice of MBE in the embedded systems domain. The paper aims to answer **RQ1**, "To what extent is MBE used in the automotive domain and how is it perceived?". It does so by focusing on the following two sub-questions.

RQ1.1: What is the current state of practice and the assessment of MBE in the embedded systems domain?

RQ1.2: How does the use and the assessment of MBE differ among different demographic subgroups in the embedded systems domain?

The paper lays the foundation for this thesis by providing empirical support that models are indeed used in the automotive domain, to which a large part of the survey participants (60) belong to. In terms of the classification introduced previously, the paper can be described as depicted in Figure 1.5. The paper does not present a defined configuration of MBRE, but rather explores which configurations exist in the automotive industry. In particular, it provides answers to the *purposes* for which models are created, *objects* which are described in models, used *notations*, *tooling*, *grade of formality* and the *point in time* at which the models are created. These aspects are marked in blue colour in Figure 1.5. Aspects that are not explicitly covered in the paper are *stakeholders*, *abstraction level* and *completeness*. Therefore, they are omitted in the figure.

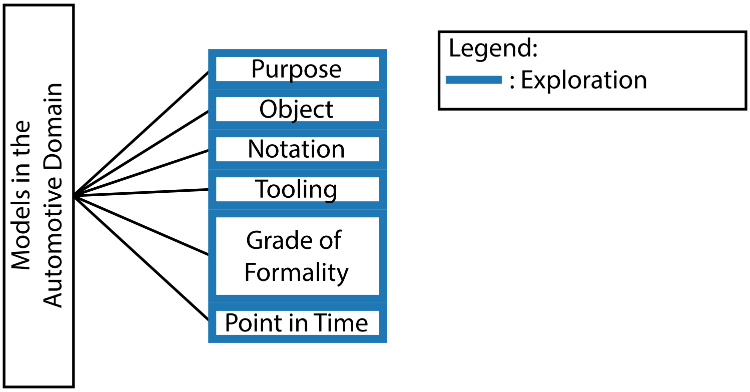

Figure 1.5: Scope of Paper A

We do not yet focus on RE specifically in this paper. Instead, we survey the entire scope of automotive systems engineering. The reason is that modelling might be common in some areas of systems engineering, but not otherwise. In this case, we can learn from the experience in those areas and try to project that experience to RE.

Using an electronic survey, we collected data from 113 individuals, mainly professionals working in the embedded systems domain. Our findings are as follows, answering **RQ1**. MBE is widely used in embedded systems and leads to a reduction of defects and improvements in quality. State machines, as used in Paper D, are one of the notations most commonly used by the participants.

Models are used for several different purposes, such as simulation or test-case generation. In particular, 49 participants use MBE during RE. We only find minor differences between subgroups, e.g., differences between the automotive and other domains. Specifically for this thesis, this means that the answers of practitioners from the automotive domain were not substantially different from practitioners in other domains.

MBE was introduced at the participants' companies for a number of *purposes*, e.g., for increased safety, traceability, quality or to shorten development time.

The participants report mainly positive effects of applying MBE, e.g., on quality or reusability. However, several shortcomings are reported as well. These include high efforts associated with education for and introduction of MBE, or increased costs.

Overall, Paper A encourages us that MBE provides the necessary benefits to address current problems in automotive RE and is at the same time accepted in the embedded domain. As a natural starting point, it triggered us to perform in-depth follow up studies on the specific situation in RE. These studies are reported in Papers B and C.

Paper A is an extended version of [71]. Furthermore, the study's results are described in a technical report [72] and were presented in [73] as an invited paper.

1.6.2 Paper B, Organisation and Communication Problems in Automotive RE

While Paper A explored the solution domain, i.e., the use of MBE in industry, Paper B explores the problem domain of automotive RE. Specifically, we explore which problems exist in automotive RE (**RQ2**). These need to be understood sufficiently well before any improvements can be made.

Existing literature on automotive RE lacks empirical support, e.g., [1,6], or aims to explain how automotive RE functions in general, e.g., [44,45]. Therefore, we aim to extract a list of problems in automotive RE. We focused on the two aspects of communication and the organisation structure, as they emerged as dominant themes during the entire data collection. Furthermore, they play a major role in the complex and distributed automotive development process. Based on this aim, the paper answers the following two research questions:

RQ2.1: What are current problems or challenges in automotive RE with respect to organisation structure and communication?

RQ2.2: How can these problems or challenges be addressed in the future?

We refer to the organisation structure as the logical relations or the "decision rule connections" between people in an organisation [74]. Communication refers to the exchange of information between individuals in an organisation or between organisations, not necessarily following the organisation structure.

As this study targets problems in RE, it does not specifically address the previously presented classification scheme for MBRE. However, it sketches possible *purposes*, *objects*, *stakeholders* and *points in time* for which models could be used in order to address the found problems.

To answer the research questions, we performed an exploratory case study, collecting data from 14 interviews at 2 automotive companies, an OEM and a supplier. From the interviews, we extracted 7 key problems related to organisation structure and communication, which we tested through a questionnaire with 31 practitioners from the automotive industry. The problems are

- P1: Lack of Product Knowledge: the lack of sufficient knowledge about the product in early stages;
- P2: Lack of Context Knowledge: the lack of context information regarding requirements on low levels of abstraction;
- P3: Unconnected Abstraction Levels: a mismatch between requirements on different abstraction levels;
- P4: Insufficient Communication and Feedback Channels: lacking communication with other people within or across the organisation;
- P5: Lack of Common Interdisciplinary Understanding: the lack of common understanding across multiple disciplines;
- P6: Unclear Responsibilities and Borders: the lack of clear and communicated responsibilities between different parts of the organisation; and
- P7: Insufficient Resources for Understanding and Maintaining Requirements: to lack enough resources in early phases to get an understanding of the needs and to maintain requirements later on.

The problems serve as a basis for solution attempts that could be targeted in the future. As a part of this, we propose a solution candidate to Problems P2, P4, and P6 in Paper E, taking into account P7.

In particular, our found problems also visualise that the complexity in automotive systems engineering is not restricted to the end-product, but rather results in a complicated organisation structure that, in turn, complicates the exchange of information. We therefore see the *need for an organisation structure that effectively supports interdisciplinary RE, taking into account the central role of software*. Here, models could take the role of a common specification language across disciplines, an aspect that is also mentioned by practitioners in Paper C.

However, one important aspect Paper B shows is that even textual specifications become extremely difficult to document and maintain in complex environments. Using purely model-based specifications could address this difficulty, if model transformations allow for different analyses, e.g., automated consistency checks. However, the sheer complexity might also render it impossible to create and maintain any kind of model-based specification that allows for such analyses. This difficulty to create a model is an issue that also emerges in Paper D.

1.6.3 Paper C, Models during RE

Papers A and B explore the problem and the solution domain in isolation. Paper C then investigates the combination of the two, namely how models can be used in the scope of automotive RE to address existing issues. That is, we investigate **RQ3**, “What is the current state and the potential of using models during automotive RE?”.

In existing literature, the use of models during RE is typically assumed to be beneficial, based on authors’ experience or anecdotal evidence. Based on these assumptions, models are then proposed for different artefacts within RE, different aspects, e.g., structure and behaviour, or different abstraction levels. Examples of such modelling frameworks include SPES 2020 [15] and REMsES [6].

In Paper C, we try to increase the empirical body of knowledge in MBRE by answering the following three research questions:

RQ3.1: How and why are models currently used during automotive RE?

RQ3.2: What is the perceived potential of models in automotive RE?

RQ3.3: Why are models not or not yet used during automotive RE?

Similarly to Papers A and B, the paper does not present one single configuration of MBRE, but explores solution candidates. That is, it outlines possible configurations of the entire feature model, in terms of *purposes*, *objects*, *stakeholders*, *notations*, *tools*, *abstraction levels*, *grade of formality*, *points in time*, and *completeness*.

To answer the research questions, we analysed the same qualitative interview data as for Paper B, but with a focus on modelling. Our findings are that models are already widely used in automotive RE, but are often informal or semi-formal. Purposes include supporting communication and understanding of requirements, and facilitating their implementation and verification later on.

Regarding the potential of modelling during RE, some interviewees consider it beneficial to have formal models for purposes such as requirements validation or verification, but others do not believe that formalisation to a high degree is feasible. Instead, the latter group would prefer informal sketches for communication purposes only. We do not see any clear distinguishing patterns in what kind of people prefer which approach. One important purpose raised by several interviewees is the use of modelling notations as a standardised vocabulary enabling cross-disciplinary discussions.

In line with findings from related work and Paper A, tools are mentioned regularly as an obstacle in practice. However, our practitioners rarely mention limited functionality with respect to modelling as an issue, but rather topics such as tool usability or interoperability.

Our findings from Paper C raise several interesting discussion topics. We find that an exclusive focus on formal models does not align with the industry needs our practitioners outline. Instead, they advocate selective use of models for specific purposes such as communication. One interesting topic is the accessibility of models by domain experts. While our interviewees do not see the actual creation of models as problematic, they highlight that engineers who need to read and understand requirements models are typically not modelling experts. That is, information stored in models needs to be made accessible in an easy and efficient manner.

1.6.4 Paper D, Exploring Modelling of Behaviour Requirements

One of the most common suggestions for MBRE is to express the requirements themselves as models, i.e., the model *objects* are requirements. However, a widespread adoption of these frameworks has not taken place in industry. Due to the introduction of new standards such as ISO26262 in the automotive industry, there has been a renewed interest in MBRE, as seen with joint industry-academia projects targeting MBRE, such as CESAR [75] or CRYSTAL [76]. In Paper D, we report on the design of a model of behavioural requirements which we created from an industrial requirements specification. That is, we aim to answer **RQ4**, "To what extent can real-life textual specifications of behaviour be translated to formal models?". This research question is broken down into two sub-questions.

RQ4.1: How can real-world requirements be formalised?

RQ4.2: How do practitioners perceive formal requirements models?

In terms of the classification introduced previously, the paper can be described as depicted in Figure 1.6. Here, we chose specific values for several of the aspects in the feature model (marked green in the figure). The model *object* represents the requirements of a safety-critical function, which we created based on an existing textual specification over the course of 2.5 years. We use the *formal* modelling *notation* timed automata (TA) [77], as the requirements were already roughly expressed in terms of states and transitions. The used modelling *tool* is Uppaal, chosen based on its availability and possibilities to perform simulations.

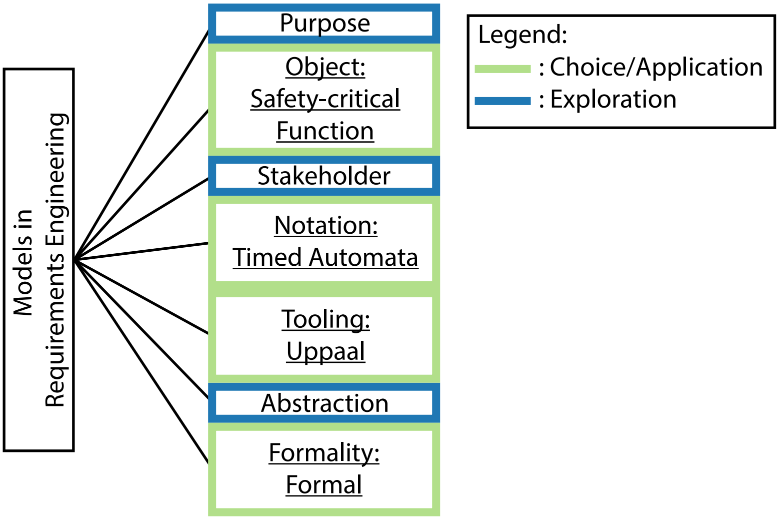

Figure 1.6: Scope of Paper D

The *purpose*, *stakeholders* and the *abstraction* were subject to exploration (marked blue in the figure). That is, we created three different model versions on different abstraction levels. The first model was a direct translation of the textual requirements, mainly aimed to form a model representation on the same level of abstraction as the textual requirements. Additionally, we wanted to be able to simulate the model. As this was not possible due to missing environment information, we created a second model on a higher level of abstraction, for simulation. Finally, we created a third, high-level model aimed to serve as an extremely simplified version of the original requirements in order to explain the function to novices.

Additionally, we generated test cases using mutation-based test case generation from the final, most abstract model. In interviews with three practitioners in the area of verification, we evaluated the outcome, as well as the feasibility and obstacles of applying the modelling approach in industry.

Our results show that behavioural requirements can be expressed as formal models. However, creating a model on the lowest level of abstraction is hardly feasible due to the high complexity and a lack of environment information. In contrast, abstraction from the detailed requirements is beneficial for both understanding and verification activities. Many of the well-known challenges in MBE were confirmed, e.g., a lack of industrial-grade tools or tool interoperability. Surprisingly, our interviewees did not consider training/education effort as a major challenge. This also reflects the findings in Paper C.

Paper D is an extended version of [78].

1.6.5 Paper E, Supporting Communication Using Existing Requirements Models

As shown in Paper B, communication during RE and of requirements knowledge is problematic in practice. Hence, methods to bridge communication gaps, in particular across organisation boundaries, are necessary. In Paper E, we propose such a method, called LoCo CoCo. We aim to answer **RQ5**, "How can existing requirements specifications be exploited for improving communication (regarding requirements)?". This question is broken down into the following four research questions.

RQ5.1: To what extent can social networks be constructed automatically from model-based systems engineering data?

RQ5.2: How do practitioners evaluate the potential of these networks to tackle known communication challenges in systems engineering?

RQ5.3: For which additional use cases would practitioners like to use LoCo CoCo?

RQ5.4: To what extent can LoCo CoCo be used with different tools and organisational contexts?

In contrast to Paper D, where we model the requirements themselves, we use in Paper E a slightly different angle towards MBRE. That is, we consider existing (textual) requirements, their connections, and connections with other software engineering artefacts as a structural model. Here, the model *object* is the entire requirements specification and the relationships (traces) between requirements. Single entities in this structural model are the individual requirements, while trace links between requirements form the associations. Using ownership and change information for each requirement, we build in this way a structural network of people and relationships between them, a *social network*. The social network is depicted using a graph *notation*, for the *purpose* of identifying the right person to communicate regarding specific issues, e.g., for requirements clarification. The resulting model *abstracts* from the actual requirements and depicts only people and their connections. This scope is depicted according to the presented MBRE classification scheme in Figure 1.7. Equivalent to Figure 1.6, aspects we chose are marked green, while aspects that we explored freely are marked blue.

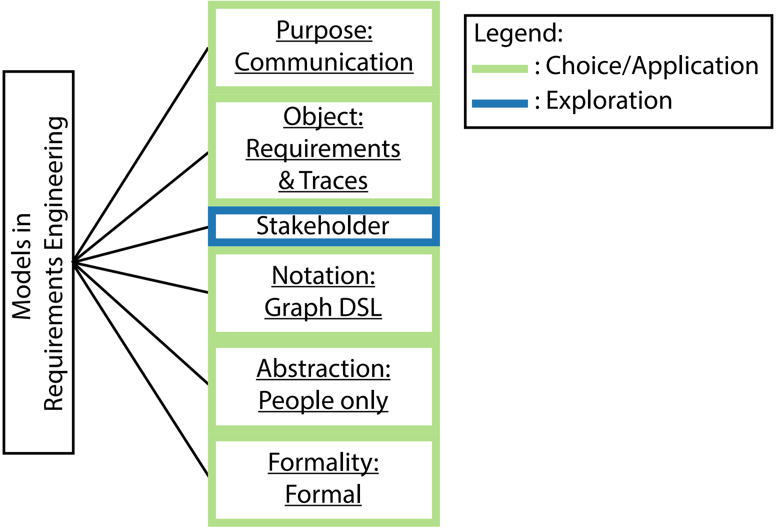

Figure 1.7: Scope of Paper E

To answer the research questions, we followed a design science research method in three iterations. In each step, we evaluated the outcome with practitioners from the case company, a Swedish automotive OEM. In total, we conducted 15 interviews and obtained 12 answers from a web-based questionnaire.

Our results indicate that LoCo CoCo can help to address existing communication challenges by identifying important contacts across the organisation structure. We observe that the quality of social data, i.e., ownership and change information related to engineers, is often low in existing systems engineering tools. However, practitioners indicate that it was sufficient, and that it could indeed trigger them to update the data in the respective tools.

Paper E does not only show the proposed approach for improving communication. Additionally, it demonstrates that the term MBRE can be used in a wider sense to promote the use of modelling techniques and understand existing techniques using MBE knowledge. For instance, mining existing data repositories, as demonstrated in Paper E, can be understood as a model transformation. This helps abstracting from the concrete implementation, omitting implementation details not relevant to the understanding of the concept.

1.7 Summary of Results

The contributions of this thesis can be summarised as follows.

The first goal of the thesis, **G1**, was to empirically investigate the current and potential use of MBE during automotive RE. In the form of our first three research questions, **RQ1** to **RQ3**, we explored this empirical basis of MBRE. The results addressing this goal are summarised in Figure 1.8.

MBE is widespread in the automotive domain and used for numerous purposes, such as simulation or test-case generation (**RQ1**). Also, the benefits of MBE are clearly seen by practitioners, in particular with respect to defect reductions and quality improvements. This indicates that it is indeed feasible and beneficial to use MBE during RE as well. However, several shortcomings also exist, e.g., a lack of tool interoperability or large required efforts to educate engineers in model-based techniques.

As a part of overall problems in automotive RE, we studied those related to communication and organisation structure (**RQ2**), as those were raised by interviewees continuously and independently of each other. Found problems include the lack of sufficient knowledge about the product in early stages or the lack of common understanding across multiple disciplines in automotive RE.

Our findings for **RQ3** show that, to some extent, models are already used in our case companies during RE. However, these models are often not of formal nature, but rather sketches of desired functionality. Similarly, meta models are used heavily to prescribe a data structure, similar to how automotive modelling standards such as AUTOSAR and EAST-ADL prescribe the structure of requirements and other system engineering artefacts. Interestingly, we find that our interviewees are neither against the use of formal models during RE, nor do they believe that a high amount of formal models is feasible. Instead, they do see the use of models for limited parts of RE, e.g., to support verification later on or to improve communication based on informal sketches. They also state that it is not realistic to introduce, and in particular to maintain, large amounts of formal specifications.

The second goal of the thesis, **G2**, was to explore solutions to existing RE challenges, motivated by **G1**, making use of models. We addressed this goal with two research questions, **RQ4** and **RQ5**. The results addressing this goal are summarised in Figure 1.9.

Towards the direction of using formal models, we aim with Paper D to explore modelling of behavioural requirements (**RQ4**). We find that even a subset of an entire requirements specification, in this case a 50-page specification for one automotive function, is difficult to express formally. The reason for this complexity is mainly the low level of detail and missing environment information. Perhaps not surprisingly, it was substantially easier to express an abstract version of the detailed behaviour requirements as a model that could then be executed and used for test-case generation. Indeed, this more abstract model was judged to be more helpful than the model representing the detailed requirements by our interviewees.

In contrast to our exploration of formal models of requirements for **RQ4**, we try to answer with **RQ5** how models can be used in different ways (i.e., not expressing the requirements themselves as models). That is, we use existing requirements data and interpret its structure as a model. In this case, the

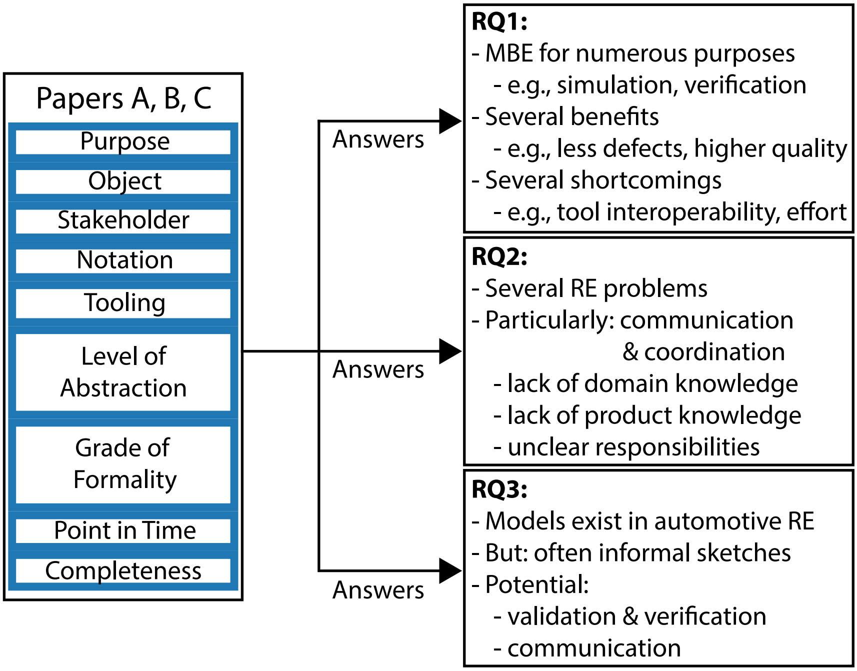
Papers A, B, C
Purpose
Object
Stakeholder
Notation
Tooling
Level of Abstraction
Grade of Formality
Point in Time
Completeness
Answers
Answers
Answers
RQ1:
- MBE for numerous purposes
- e.g., simulation, verification
- Several benefits
- e.g., less defects, higher quality
- Several shortcomings
- e.g., tool interoperability, effort
RQ2:
- Several RE problems
- Particularly: communication & coordination
- lack of domain knowledge
- lack of product knowledge
- unclear responsibilities
RQ3:
- Models exist in automotive RE
- But: often informal sketches
- Potential:
- validation & verification
- communication

Figure 1.8: Summary of Research Questions Addressing Goal G1

model *object* is the structure of the requirements specification instead of the requirements themselves. The advantage of this approach is that there is no formal modelling knowledge required by engineers using this data. Instead, we can exploit existing data to help address issues we found for **RQ2**, e.g., a lack of communication and coordination. Our findings show that it is indeed possible to aid engineers in finding the right people for requirements clarification, even though existing data is in practice often incomplete or incorrect.

1.8 Discussion

The joint outcomes of the five papers included in this thesis raise a number of important discussion points regarding MBRE, both in the automotive domain and in general. First, the essential discussion point is if models are indeed the right way to address existing problems in RE. Secondly, given that models are the desired way forward, it needs to be discussed what the sweet spot of MBRE is. The third discussion point is the choice of strategy to introduce MBRE in an organisation. Finally, the outcome of this thesis raises a number of discussion points regarding the overall vision to use model-based techniques in SE. We discuss these four points in the following.

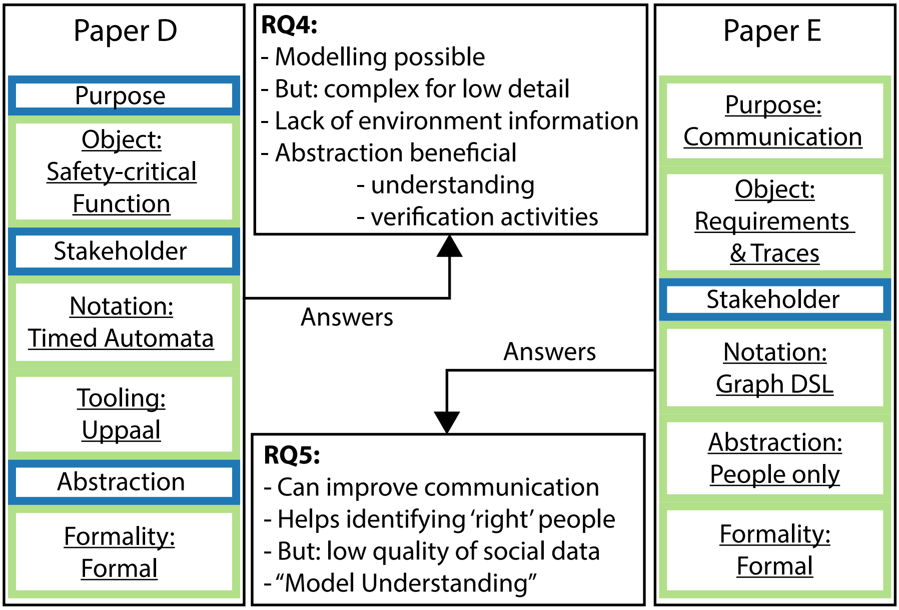

Figure 1.9: Summary of Research Questions Addressing Goal G2

1.8.1 Modelling or not?

The problems found in Paper B raise several needs for future research, e.g., the need to effectively support interdisciplinary RE and communication. Models were raised by interviewees in Paper C as one possible means to address these needs. Additionally, we demonstrate the feasibility to do so for a specific case in Paper E. However, instead of using formal models, some practitioners also suggested lowering the overall effort for RE. That is, we see two contrasting approaches: to impose stricter rules or a more formal process, and to relax the existing processes and lower the overall effort spent for RE-related activities. The latter approach is typically advocated for together with the use or introduction of agile methods. That is, striving for lowering documentation in line with the agile manifesto [79].

Going beyond RE, it is not uncommon in industry and academia to view agile methods as a counterpart to model-centric methods, or at least see those two worlds in conflict with each other. However, in a recent study on RE in large-scale agile environments, we also find that practitioners suggest the use of models to maintain an understanding of the requirements in agile processes [80]. Similarly, Sikora et al. [56] report that practitioners would like to use more models. Finally, many of the suggestions in Paper C, as well as the high-abstraction model in Paper D could be used together with agile methods in an iterative fashion. Hence, the topic of modelling in RE remains timely, even with the advent of agile methods.

1.8.2 Where is the Modelling Sweet Spot?

While a broad use of MBE would theoretically maximise the benefits of using models during RE, our data does not anyhow support that this is feasible or desirable. Practitioners lean much more towards a targeted introduction of MBE for certain concerns only. While most of the interviewees would wish for a broader use of models, none expressed the desire for a proper model-driven process in which models are indeed the main artefacts. Hence, there is an interesting trade-off between being as formal as possible, to maximise the benefits obtained from MBE, and restricting the amount of formal models, so that the creation and maintenance effort does not become overwhelming.

As shown in Paper D, the complexity of models seems to limit the applicability of requirements modelling on a low level of detail, something that is corroborated by most practitioners in Paper C. While a deeper domain knowledge could have simplified the modelling process, it would likely still remain cumbersome. In particular, while requirements engineers can be expected to have much more domain knowledge, it can also be expected that they have substantially less knowledge in modelling than we do. Hence, complexity might be a limiting factor.

Overall, the suggestion to use models for a few specific purposes during RE is in contrast to both research that promotes heavy use of models during RE and other parts of a typical SE process, e.g., [6, 15], and research reporting the practitioner opinion that formal models are too much effort given their benefits, e.g., [41, 42]. This makes for interesting future work investigating the sweet spot of MBRE.

1.8.3 How to Introduce Modelling?

Throughout our data analyses, the question of how models and modelling can be introduced, be it during or outside RE, comes up. Both in Papers C and D, interviewees state that it is difficult to convince management of introducing modelling, as it is hard to directly show the positive effects of MBE. Similarly, the argument that there is not enough time to re-visit existing requirements and convert them to formal models (or even just to re-write them textually) is raised. This is in line with the findings by Hutchinson et al. [10], who report that introducing MBE requires an iterative approach. Similarly, Stieglbauer [81] argues for so-called *Micro Injections*, introducing MBE in small iterations with a clear, demonstratable outcome.

In the context of introducing modelling, individual roles and preferences are also commonly mentioned. That is, several interviewees raise the concern that individuals might resist the introduction of modelling, as it would potentially challenge their own role or position as a distinguished expert in a certain field. Similarly, this point is found in related literature on MBE, e.g., in [11]. While resistance to change is natural, personality might play an important role. For instance, Whittle et al. [12, 82] report that software architects were more likely to embrace MDE in comparison to programmers. That is, the success of model-centric techniques depends on the individuals and how they embrace change. Given these observed personality differences, something that we recognised also in modelling education [83], we are currently investigating to what extent personal preferences for abstraction or detail influence one's own view on modelling. This is in line with existing research on personality types in SE, e.g., [84–86]. Given that we can indeed observe differences in the perception of modelling with respect to the personality of engineers, this would also impact introduction strategies for MBE, and the use of MBE in general.

Finally, substantial shortcomings with MBE remain and need to be addressed if MBE should play an important role in industry in the future. Several of these are related to the effort required to introduce or use MBE, and the complexity to maintain large amounts of model-based artefacts. While related work, e.g., [11, 26], typically raises training effort in MBE as one of the main shortcomings of MBE, our findings are contradictory in this point. While in Paper A, a large amount of survey participants agrees that there is a high training effort in order to be able to use MBE, several of the interviewees in both Paper C and D do not voice any concerns with respect to modelling training. In most cases, our interviewees come from an engineering background in either electrical or mechanical engineering. This could have affected their view on formal modelling techniques, as modelling is common in both disciplines. In fact, Hutchinson et al. [11] also find that companies with SE as their main business find MBE harder to adopt than other companies. Hence, this is a topic that needs to be investigated in more detail in order to tailor MBRE and MBE in general, and their introduction, to the specific organisation context.

1.8.4 What is Beyond the MDE Vision?

In the traditional sense of MDE and MDA, models are created to reflect different aspects of the system, transformed into more refined models and finally into code. However, several of the findings in this thesis question this vision.

First, we clearly see the value of abstraction and reduction in Paper D. Our final, abstract model lacks important information. It does not only abstract from implementation details as a platform-independent model in MDA would, it actually misses relevant functionality, e.g., connections to other sub-systems. Therefore, using this model as one puzzle piece in a transformation chain towards a final, running system is unlikely. That is, program code is not the aim of the model we created in Paper D. Instead, it clearly is helpful for understanding purposes and even for formal techniques such as test-case generation. In summary, the use of models can be beneficial even if used for single concerns, not addressing an overall model-driven vision.

Similarly, Paper E shows an example of using the MBE knowledge and vocabulary in the context of a repository mining task. That is, the idea of MBE is not targeted at actively creating a model that reflects parts of the system, but rather understanding existing information as a model. Similarly to how other disciplines use models, as a means to better understand a complex problem, SE could benefit from using existing MBE knowledge as a toolset to abstract from detail in existing approaches.

Overall, these two points motivate the use of MBE in SE, going beyond the traditional MDE vision. In fact, in recent years, several of the keynotes at the MODELS conference, the premier academic conference for modelling topics in SE, have motivated the use of MBE knowledge in fields outside of SE, e.g., [87, 88].

1.9 Validity Threats

In this section, we give an overview of the threats to the validity of the results of this thesis, structured using the classification into construct, internal, external validity and reliability as given in [64]. Detailed threats to validity of the included publications are discussed in their respective chapters, i.e., Chapters 2 to 6.

1.9.1 Construct Validity

Construct validity reflects the extent to which the studied factors "really represent what the researcher has in mind and what is investigated according to the research questions" [64]. For instance, terms and definitions might be interpreted in different ways by the researcher and interviewees, leading to a threat to construct validity.

To increase the construct validity, we used pilots and/or reviews of our data collection guides in all studies included in this thesis. That is, we tested interview guides with practitioners before collecting data, sent out questionnaires for review, both by practitioners and researchers, and finally discussed the guides internally. In particular, we checked for and tried to avoid leading questions or any (implicit) hypotheses being included in the instruments.

Despite the reviews of our data collection instruments, certain threats to construct validity remain. For example, we see the threat that the essential terms used in this thesis, *requirement* and *model*, are used rather differently by different people. While we use the IREB definition in which a requirement is a stakeholder need or a system capability [20], practitioners might interpret the term more freely. In particular, it is commonly discouraged to include design details in requirements, e.g., by [22, 35]. In practice, however, due to the large amount of different supplier and teams working on an automotive system, automotive requirements contain large amounts of detailed design, e.g., interface descriptions. Similarly, a *model* is, according to the definition, essentially an abstraction from reality. In practice, some people might only consider graphical models as models, e.g., UML models, while others would be much more inclusive, e.g., consider even program code to be a model. We therefore tried to interpret the terms rather freely in our data collection and discussed them with the study participants when needed.

1.9.2 Internal Validity

Internal validity concerns the validity of the examined causal relations in a study [64]. If factors that affect each other are investigated, there is the risk that other, unknown factors also affect the outcome.

Similarly to construct validity, we tried to examine, through reviews and pilots, whether there are any factors left out in the data collection instruments. Furthermore, we asked open-ended questions in both interviews and surveys to ensure that participants could voice their opinion regarding any other issues that might affect our study topic.

Still, internal validity remains a concern, in particular for case studies reported in Papers B and C. In this case, we study a topic in its real-life context. As such, there are many complex interactions of factors affecting the study topic. For instance, the problems reported in Paper B might have other, underlying root causes, as for example the ones proposed by Bjarnason et al. in [46]. Given the current lack of general theories in SE and a lack of substantial amounts of data on the topic, we did not attempt to perform any kind of deeper analyses on their relation.

1.9.3 External Validity

External validity describes the amount to which findings can be generalised to situations outside of the study context [64]. The external validity of case studies is generally low, as they study a concept in its context. However, the intention for case studies is still to “enable analytical generalization where the results are extended to cases which have common characteristics and hence for which the findings are relevant, that is, defining a theory.” [64].

For the two case studies included in this thesis, as well as the design science studies, the external validity is low by design. Findings are to a large extent specific to the study context. This is a threat we accept, as we deemed the goal to obtain more in-depth data on the topic of MBRE more important than attempting to find general, but more vague answers. However, we used different strategies to increase the generalisability of our findings, at least to other automotive companies. For example, we sampled our interviewees from different roles and with different experience, thus broadening the context which we investigate. In Paper B, we conducted a follow-up survey to evaluate the findings from the study context with a broader sample of participants from other automotive companies.

Paper A has a high external validity, given the survey research strategy. As a limitation, it has to be noted that our survey sample included almost exclusively proponents of MBE. That is, we cannot make any conclusions regarding the use of MBE in the entire embedded systems industry, but rather from a point of view of MBE proponents. This could lead to an overemphasis on the positive evaluation of MBE, as proponents could be biased towards the positive effects of MBE, but also gives interesting insights regarding the perceived negative effects of MBE.

1.9.4 Reliability

Reliability describes to what extent other researchers would reach the same conclusions if they were to repeat the study under the same conditions and following the same method [64].

For all five included papers, we published the data collection instruments and described the data analysis in detail. Where possible, we also published the raw data on which we based our analyses. In the case of interviews, we were not able to publish the raw transcripts due to non-disclosure agreements. Still, reliability should be at a reasonable level compared to the current methodological state in SE.

As an exception, the study presented in Paper D has larger threats to reliability. In this case, the evaluation procedures should be reliable, as both simulation and test-case generation were to a large extent an automated process, and the interview guide has been published. However, model creation is a process that was heavily dependent on the knowledge and understanding of the researcher who created the models. We did not attempt a replication in which other researchers or practitioners would create the same models, which means that we do not have any findings regarding the reliability of this process. Furthermore, we performed empirical evaluation with 3 interviewees only. Therefore, Paper D can be seen as an exploration that needs further follow-up studies to reach a high reliability.

1.10 Conclusions and Future Work

With this thesis, we contribute to the body of knowledge in automotive MBRE. We do so by following a broad, empirical approach, providing an overview of the state of practice in automotive MBE (**RQ1**), the potential of MBRE in the automotive domain (**RQ3**), as well as problems currently faced in automotive RE (**RQ2**).

Generally, MBE is seen as potentially beneficial by practitioners, but a number of shortcomings need to be addressed. Additionally, practitioners question the feasibility of introducing MBRE broadly, covering many different artefacts and abstraction levels. This raises the question where the sweet spot lies for MBRE.

The current use of models during RE is often limited to informal or semi-formal notations, but practitioners support the introduction of formal models for purposes such as improving requirements understanding or supporting verification activities. Hence, the question raised by practitioners is not whether modelling should be used or not, but rather how much modelling there should be.

Given the amount of studies that investigate different aspects of MBE adoption, we see the need for future work that reviews and synthesises this work in a systematic fashion. In this context, the aim should be to build a robust theory of MBE use and adoption in industry. In particular, there is a need to synthesise existing work on the introduction of MBE in industry.

Similarly to the topic of MBE adoption, we see the need for future work towards theory building in RE challenges. Bjarnason et al. [46] have started building a framework of root causes related to RE communication challenges. Given the challenges we encountered in the appended publications, as well as in related work, we believe that this is a worthwhile effort that should be continued.

Focusing specifically on MBRE, we see the need for more qualitative as well as quantitative studies that investigate the actual needs in more detail. In particular, judging from the limited adoption in industry, it is clearly insufficient to base future work on assumptions how models should be used and introduced. Instead, systematic empirical work is needed.

Beyond empirical studies that investigate the current state of practice, we found the iterative approach we followed in Papers D (**RQ4**) and E (**RQ5**) rewarding in the sense that it allowed a frequent exchange with and evaluation by practitioners. That is, a regular *sanity check* helps to root the proposed approaches in a real-world context. Based on this experience, we suggest similar iterative projects with industry partners, e.g., following the design science research or action research strategies. Clearly, this is only feasible for smaller studies that do not count on a broad introduction of a new technique or method in industry.

Many of the concerns we encountered throughout the appended publications relate to organisation-wide issues. In the case of the automotive industry, this includes various different disciplines and backgrounds. Therefore, there is a clear need to tackle existing problems in interdisciplinary projects. For MBRE in particular, it seems unlikely that any approach which does not cater to the needs of mechanical and electrical engineers as well would be introduced in our case companies, at least not on a broader scale.

In Section 1.9 we discussed that reliability is low for Paper D, as models were created by one researcher only. While there has been research on model patterns in the past, also for behaviour models, we currently lack systematic ways of creating abstractions and, therefore, models. Related work on (software) modelling education also clearly points to this issue, reporting difficulties in teaching students abstraction and modelling. Therefore, there is a need for future research investigating how engineers create models. That is, discovering and formalising different styles of model creation and of abstraction. Towards this direction, we are currently conducting a study in which we observe students creating models of requirements.

Finally, Paper E has shown that existing structures can be understood as a model, which helps abstracting from the concrete problem. This idea goes beyond the traditional MDE vision, in which models of the system are created with the purpose to reach a running system. Instead, we show that we can use the MBE knowledge to understand existing problems and solutions in an abstract way. In other topics that are extending into all scientific disciplines, e.g., machine learning or data science applications, the body of knowledge from MBE could help in a similar fashion, abstracting from the concrete, rather complex algorithms and applications.

Bibliography

[1] M. Broy, "Challenges in automotive software engineering," in *28th International Conference on Software Engineering (ICSE '06)*, 2006, pp. 33–42.

[2] C. Ebert and C. Jones, "Embedded software: Facts, figures, and future," *Computer*, vol. 42, no. 4, pp. 42–52, 2009.

[3] J. D. Procaccino, J. M. Verner, S. P. Overmyer, and M. E. Darter, "Case study: factors for early prediction of software development success," *Information and Software Technology*, vol. 44, no. 1, pp. 53–62, 2002.

[4] J. S. Reel, "Critical success factors in software projects," *IEEE Software*, vol. 16, no. 3, pp. 18–23, 1999.

[5] N. Cerpa and J. M. Verner, "Why did your project fail?" *Commun. ACM*, vol. 52, no. 12, pp. 130–134, 2009.

[6] P. Braun, M. Broy, F. Houdek, M. Kirchmayr, M. Müller, B. Penzenstadler, K. Pohl, and T. Weyer, "Guiding requirements engineering for software-intensive embedded systems in the automotive industry," *Computer Science - Research and Development*, vol. 29, no. 1, pp. 21–43, 2014.

[7] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, ser. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2012.

[8] P. Baker, S. Loh, and F. Weil, "Model-driven engineering in a large industrial context - motorola case study," in *ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems (MODELS '05)*, 2005, pp. 476–491.

[9] P. Mohagheghi, W. Gilani, A. Stefanescu, M. A. Fernandez, B. Nordmoen, and M. Fritzsche, "Where does model-driven engineering help? experiences from three industrial cases," *Software and Systems Modeling*, vol. 12, no. 3, pp. 619–639, 2013.

[10] J. Hutchinson, M. Rouncefield, and J. Whittle, "Model-driven engineering practices in industry," in *33rd International Conference on Software Engineering (ICSE '11)*, 2011, pp. 633–642.

[11] J. Hutchinson, J. Whittle, M. Rouncefield, and S. Kristoffersen, "Empirical assessment of MDE in industry," in *33rd International Conference on Software Engineering (ICSE '11)*, 2011, pp. 471–480.

[12] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, "Industrial adoption of model-driven engineering: Are the tools really the problem?" in *ACM/IEEE 16th International Conference on Model Driven Engineering Language and Systems (MODELS '13)*, 2013, pp. 1–17.

[13] J. P. Bowen, R. W. Butler, D. L. Dill, R. L. Glass, D. Gries, and A. Hall, "An invitation to formal methods," *Computer*, vol. 29, no. 4, pp. 16–, 1996.

[14] M. Jackson, "Formalism and informality in RE," in *5th IEEE International Symposium on Requirements Engineering (RE '11)*, 2001, pp. 269–269.

[15] K. Pohl, H. Hönninger, R. Achatz, and M. Broy, *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology.* Springer Science & Business Media, 2012.

[16] M. Maurer and H. Winner, *Automotive systems engineering.* Springer, 2013.

[17] K. Niesel, "Volvo Car Group, Software center Day Skåne," last accessed Feb. 2018. [Online]. Available: http://www.software-center.se/digitalAssets/1506/1506474_sw-center---sk--ne-volvo.pdf

[18] D. Damian and J. Chisan, "An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management," *Transactions on Software Engineering, IEEE*, vol. 32, no. 7, pp. 433–453, 2006.

[19] D. Damian, J. Chisan, L. Vaidyanathasamy, and Y. Pal, "Requirements engineering and downstream software development: Findings from a case study," *Empirical Software Engineering*, vol. 10, no. 3, pp. 255–283, 2005.

[20] M. Glinz, "A glossary of requirements engineering terminology," *Standard Glossary of the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version*, vol. 1, 2011.

[21] "Ieee recommended practice for software requirements specifications," *IEEE Std 830-1998*, pp. 1–40, 1998.

[22] S. Lauesen, *Software Requirements: Styles and Techniques*, 1st ed. Pearson Education, 2001.

[23] H. Stachowiak, Ed., *Allgemeine Modelltheorie.* Springer, 1973.

[24] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *Future of Software Engineering (FOSE '07)*, 2007, pp. 37–54.

[25] P. Mohagheghi and V. Dehlen, “Where is the proof? - a review of experiences from applying MDE in industry,” in *Model Driven Architecture - Foundations and Applications*, ser. Lecture Notes in Computer Science, I. Schieferdecker and A. Hartman, Eds., 2008, vol. 5095, pp. 432–443.

[26] L. T. W. Agner, I. W. Soares, P. C. Stadzisz, and J. M. Simão, “A brazilian survey on UML and model-driven practices for embedded software development,” *Journal of Systems and Software*, vol. 86, no. 4, pp. 997–1005, 2013, SI : Software Engineering in Brazil: Retrospective and Prospective Views.

[27] R. Soley *et al.*, “Model driven architecture,” *OMG white paper*, vol. 308, no. 308, p. 5, 2000.

[28] J. Bézivin, “In search of a basic principle for model driven engineering,” *Novatica Journal, Special Issue*, vol. 5, no. 2, pp. 21–24, 2004.

[29] Object Management Group, “Unified modeling language,” last accessed Feb. 2018. [Online]. Available: http://www.omg.org/spec/UML

[30] ——, “Overview and guide to omg’s architecture,” last accessed Jan. 2016. [Online]. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01

[31] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev, “Atl: A model transformation tool,” *Science of Computer Programming*, vol. 72, no. 1-2, pp. 31–39, 2008, special Issue on Second issue of experimental software and toolkits (EST).

[32] Object Management Group, “Meta Object Facility (MOF) 2.0 Query/View/Transformation (QVT), v1.2,” last accessed Jun. 2014. [Online]. Available: http://www.omg.org/spec/QVT/1.2/

[33] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, “Feature-oriented domain analysis (foda) feasibility study,” DTIC Document, Tech. Rep., 1990.

[34] E. S.-K. Yu, “Modelling strategic relationships for process reengineering,” Ph.D. dissertation, 1996, uMI Order No. GAXNN-02887 (Canadian dissertation).

[35] T. Gorschek and C. Wohlin, “Requirements abstraction model,” *Requirements Engineering*, vol. 11, no. 1, pp. 79–101, 2006.

[36] M. Famelis, S. Ben-David, M. Chechik, and R. Salay, “Partial models: A position paper,” in *8th International Workshop on Model-Driven Engineering, Verification and Validation (MoDeVVa ’11)*, 2011, pp. 1:1–1:4.

[37] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, “A taxonomy of tool-related issues affecting the adoption of model-driven engineering,” *Software & Systems Modeling*, vol. 16, no. 2, pp. 313–331, 2017.

[38] S. Kirstan and J. Zimmermann, "Evaluating costs and benefits of model-based development of embedded software systems in the car industry–results of a qualitative case study," in *Workshop C2M: EEMDD "From code centric to model centric: Evaluating the effectiveness of MDD"*, 2010.

[39] A. Forward and T. C. Lethbridge, "Problems and opportunities for model-centric versus code-centric software development: A survey of software professionals," in *International Workshop on Models in Software Engineering (MiSE '08)*, 2008, pp. 27–32.

[40] J. Hutchinson, J. Whittle, and M. Rouncefield, "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure," *Science of Computer Programming*, vol. 89, Part B, no. 0, pp. 144–161, September 2014, SI: Success Stories in Model Driven Engineering.

[41] M. Torchiano, F. Tomassetti, F. Ricca, A. Tiso, and G. Reggio, "Preliminary findings from a survey on the md* state of the practice," in *5th International Symposium on Empirical Software Engineering and Measurement (ESEM '11)*, 2011, pp. 372–375.

[42] T. Gorschek, E. Tempero, and L. Angelis, "On the use of software design models in software development practice: An empirical investigation," *Journal of Systems and Software*, vol. 95, pp. 176–193, 2014.

[43] R. Heldal, P. Pelliccione, U. Eliasson, J. Lantz, J. Derehag, and J. Whittle, "Descriptive vs prescriptive models in industry," in *ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, 2016, pp. 216–226.

[44] J. Pernstål, A. Magazinius, and T. Gorschek, "A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 22, no. 7, pp. 965–1004, 2012.

[45] L. Almefelt, F. Berglund, P. Nilsson, and J. Malmqvist, "Requirements management in practice: findings from an empirical study in the automotive industry," *Research in Engineering Design*, vol. 17, no. 3, pp. 113–134, 2006.

[46] 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 *19th IEEE International Requirements Engineering Conference (RE '11)*, 2011, pp. 37–46.

[47] L. Karlsson, Å. G. Dahlstedt, B. Regnell, J. Natt och Dag, and A. Persson, "Requirements engineering challenges in market-driven software development - an interview study with practitioners," *Information and Software Technology*, vol. 49, no. 6, pp. 588–604, 2007.

[48] S. Morris, "Models in the RE series," in *21st IEEE International Requirements Engineering Conference (RE '13)*, 2013, pp. 230–237.

[49] A. Pretschner, M. Broy, I. H. Kruger, and T. Stauner, "Software engineering for automotive systems: A roadmap," in *Future of Software Engineering (FOSE '07)*, 2007, pp. 55–71.

[50] EAST-ADL Association, "EAST-ADL v2.1.12," last accessed Jun. 2015. [Online]. Available: http://www.east-adl.info/Specification/V2.1.12/html/index.html

[51] Object Management Group, "SysML 1.3," last accessed Jun. 2015. [Online]. Available: http://www.omg.org/spec/SysML/1.3/

[52] J.-L. Boulanger and Q. D. Văn, "Requirements engineering in a model-based methodology for embedded automotive software," in *IEEE International Conference on Research, Innovation and Vision for the Future (RIVF '08)*, 2008.

[53] J. Piques and E. Andrianarison, "Sysml for embedded automotive systems: lessons learned," *Interfaces*, vol. 3, p. 3b, 2011.

[54] A. Albinet, S. Begoc, J. Boulanger, O. Casse, I. Dal, H. Dubois, F. Lakhal, D. Louar, M. Peraldi-Frati, Y. Sorel *et al.*, "The memvatex methodology: from requirements to models in automotive application design," in *4th European Congress on Embedded Real Time Software (ERTS '08)*, 2008.

[55] M. Lubars, C. Potts, and C. Richter, "A review of the state of the practice in requirements modeling," in *IEEE International Symposium on Requirements Engineering (RE '93)*, 1993, pp. 2–14.

[56] E. Sikora, B. Tenbergen, and K. Pohl, "Requirements engineering for embedded systems: An investigation of industry needs," in *Requirements Engineering: Foundation for Software Quality*, ser. Lecture Notes in Computer Science, D. Berry and X. Franch, Eds., 2011, vol. 6606, pp. 151–165.

[57] D. Méndez Fernández, S. Wagner, K. Lochmann, A. Baumann, and H. de Carne, "Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies," *Information and Software Technology*, vol. 54, no. 2, 2011.

[58] W. Böhm, S. Henkler, F. Houdek, A. Vogelsang, and T. Weyer, "Bridging the gap between systems and software engineering by using the spes modeling framework as a general systems engineering philosophy," *Procedia Computer Science*, vol. 28, pp. 187–194, 2014.

[59] W. Böhm, M. Junker, A. Vogelsang, S. Teufl, R. Pinger, and K. Rahn, "A formal systems engineering approach in practice: An experience report," in *1st International Workshop on Software Engineering Research and Industrial Practices*, 2014, pp. 34–41.

[60] J. Brings, J. Bellendorf, K. Keller, M. Kempe, N. Kurt, A. Palm, and M. Daun, "Applying the spes modeling framework," in *REFSQ '17 Workshops, Doctoral Symposium, Research Method Track, and Poster Track*, 2017.

[61] V. Brandstetter, A. Froese, B. Tenbergen, A. Vogelsang, J. C. Wehrstedt, and T. Weyer, "Early validation of automation plant control software using simulation based on assumption modeling and validation use cases," *Complex Systems Informatics and Modeling Quarterly*, no. 4, pp. 50–65, 2015.

[62] M. Fockel and J. Holtmann, "A requirements engineering methodology combining models and controlled natural language," in *IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE '14)*, 2014, pp. 67–76.

[63] Z. A. Barmi, A. H. Ebrahimi, and R. Feldt, "Alignment of requirements specification and testing: A systematic mapping study," in *4th International Conference on Software Testing, Verification and Validation Workshops (ICSTW '11)*, 2011, pp. 476–485.

[64] P. Runeson, M. Höst, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering*, 1st ed. Wiley, 2012.

[65] V. R. Basili, "The experimental paradigm in software engineering," in *Experimental Software Engineering Issues: Critical Assessment and Future Directions*, ser. Lecture Notes in Computer Science, H. D. Rombach, V. R. Basili, and R. W. Selby, Eds., 1993, vol. 706, pp. 1–12.

[66] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, and B. Regnell, *Experimentation in Software Engineering.* Springer, 2012.

[67] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, Mar. 2004.

[68] R. J. Wieringa, *Design science methodology for information systems and software engineering.* Springer, 2014.

[69] L. McLeod, S. G. MacDonell, and B. Doolin, "Qualitative research on software development: a longitudinal case study methodology," *Empirical Software Engineering*, vol. 16, no. 4, pp. 430–459, 2011.

[70] Object Management Group, "UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems," 2011, last accessed Jan. 2016. [Online]. Available: http://www.omg.org/spec/MARTE/1.1/PDF

[71] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Assessing the state-of-practice of model-based engineering in the embedded systems domain," in *ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MODELS '14)*, 2014, pp. 166–182.

[72] N. Marko, G. Liebel, D. Sauter, M. Lodwich, Aleksander Tichy, A. Leitner, and J. Hansson, "Model-based engineering for embedded systems in practice," University of Gothenburg, Tech. Rep. 2014:01, November 2014. [Online]. Available: http://hdl.handle.net/2077/37424

[73] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, “Industrielle praxis modellbasierter entwicklung im bereich eingebetteter systeme,” in *Software Engineering & Management 2015, Multikonferenz der GI-Fachbereiche Softwaretechnik (SWT) und Wirtschaftsinformatik (WI), FA WI-MAW*, 2015, pp. 37–38.

[74] H. H. Baligh, *Organization structures: theory and design, analysis and prescription*, 1st ed. Springer, 2006.

[75] CESAR Project, “CESAR - Cost-efficient methods and processes for safety relevant embedded systems,” http://www.cesarproject.eu, last accessed Mar. 2016.

[76] CRYSTAL Project, “CRYSTAL - CRitical sYSTem engineering AcceLeration,” last accessed Jan. 2016. [Online]. Available: http://www.crystal-artemis.eu

[77] R. Alur and D. L. Dill, “A Theory of Timed Automata,” *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[78] G. Liebel, A. Anjorin, E. Knauss, F. Lorber, and M. Tichy, “Modelling behavioural requirements and alignment with verification in the embedded industry.” in *5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD ’17)*, 2017, pp. 427–434.

[79] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, “Manifesto for agile software development,” 2001.

[80] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, “Requirements engineering challenges in large-scale agile system development,” in *IEEE 25th International Requirements Engineering Conference (RE ’17)*, 2017, pp. 352–361.

[81] G. Stieglbauer and I. Roncevic, “Objecting to the revolution: Model-based engineering and the industry-root causes beyond classical research topics.” in *5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD ’17)*, 2017, pp. 629–639.

[82] J. Whittle, J. Hutchinson, and M. Rouncefield, “The state of practice in model-driven engineering,” *IEEE Software*, vol. 31, no. 3, pp. 79–85, 2014.

[83] G. Liebel, R. Heldal, and J. P. Steghöfer, “Impact of the use of industrial modelling tools on modelling education,” in *IEEE 29th International Conference on Software Engineering Education and Training (CSEET ’16)*, 2016, pp. 18–27.

[84] B. Curtis, “Fifteen years of psychology in software engineering: Individual differences and cognitive science,” in *7th International Conference on Software Engineering (ICSE ’84)*, 1984, pp. 97–106.

[85] R. Feldt, L. Angelis, R. Torkar, and M. Samuelsson, “Links between the personalities, views and attitudes of software engineers,” *Information and Software Technology*, vol. 52, no. 6, pp. 611–624, 2010.

[86] M. V. Kosti, R. Feldt, and L. Angelis, “Personality, emotional intelligence and work preferences in software engineering: An empirical study,” *Information and Software Technology*, vol. 56, no. 8, pp. 973–990, 2014.

[87] S. Easterbrook, “Modelling the climate system: Is model-based science like model-based engineering? (keynote),” in *ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS ’15)*, 2015, pp. 1–1.

[88] G. Blair, “Grand challenges, grand responses? (keynote),” in *ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS ’16)*, 2016, pp. 1–1.

[89] “IEEE Standard for Information Technology- Standardized Application Environment Profile (AEP)-POSIX Realtime and Embedded Application Support,” *IEEE Std 1003.13-2003 (Revision of IEEE Std 1003.13-1998)*, pp. i – 164, 2004.

[90] G. Karsai, S. Neema, and D. Sharp, “Model-driven architecture for embedded software: A synopsis and an example,” *Science of Computer Programming*, vol. 73, no. 1, pp. 26 – 38, 2008, special Issue on Foundations and Applications of Model Driven Architecture (MDA).

[91] H. Espinoza, D. Cancila, B. Selic, and S. Gérard, “Challenges in combining sysml and marte for model-based design of embedded systems,” in *5th European Conference on Model Driven Architecture - Foundations and Applications*, 2009, pp. 98–113.

[92] M. Grossman, J. E. Aronson, and R. V. McCarthy, “Does UML make the grade? insights from the software development community,” *Information and Software Technology*, vol. 47, no. 6, pp. 383–397, 2005.

[93] B. Anda, K. Hansen, I. Gullesen, and H. Thorsen, “Experiences from introducing uml-based development in a large safety-critical project,” *Empirical Software Engineering*, vol. 11, no. 4, pp. 555–581, 2006.

[94] B. Dobing and J. Parsons, “Dimensions of uml diagram use: A survey of practitioners,” *Journal of Database Management*, vol. 19, no. 1, pp. 1–18, 2008.

[95] S. L. Pfleeger and B. A. Kitchenham, “Principles of survey research: Part 1: Turning lemons into lemonade,” *SIGSOFT Software Engingeering Notes*, vol. 26, no. 6, pp. 16–18, 2001.

[96] T. Punter, M. Ciolkowski, B. G. Freimut, and I. John, “Conducting online surveys in software engineering,” in *2003 International Symposium on Empirical Software Engineering (ISESE ’03)*, 2003.

[97] D. J. Leiner, “Too fast, too straight, too weird: Post hoc identification of meaningless data in internet surveys,” 2013, last accessed Jan. 2016. [Online]. Available: http://www.researchgate.net/publication/258997762_Too_Fast_too_Straight_too_Weird_Post_Hoc_Identification_of_Meaningless_Data_in_Internet_Surveys

[98] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, “Formal methods: Practice and experience,” *ACM Computing Surveys*, vol. 41, no. 4, pp. 19:1 – 19:36, Oct. 2009.

[99] R. A. Fisher, “On the interpretation of χ^2 from contingency tables, and the calculation of p,” *Journal of the Royal Statistical Society*, vol. 85, no. 1, pp. 87–94, 1922.

[100] J. G. March and H. A. Simon, *Organizations*, 2nd ed. Wiley, 1993.

[101] S. Chakraborty and S. Ramesh, “Guest editorial special section on automotive embedded systems and software,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1701–1703, 2015.

[102] B. Curtis, H. Krasner, and N. Iscoe, “A field study of the software design process for large systems,” *Commun. ACM*, vol. 31, no. 11, pp. 1268–1287, 1988.

[103] S. Hansen and K. Lyytinen, “Challenges in contemporary requirements practice,” in *43rd Hawaii International Conference on System Sciences (HICSS ’10)*, 2010, pp. 1–11.

[104] S. Altman, E. Valenzi, and R. M. Hodgetts, Eds., *Organizational Behavior: Theory and Practice*, 1st ed. Academic Press, 1985.

[105] INCOSE, “INCOSE - What is Systems Engineering?” last accessed Mar. 2015. [Online]. Available: http://www.incose.org/practice/whatissystemseng.aspx

[106] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, and R. Feldt, “Challenges and practices in aligning requirements with verification and validation: a case study of six companies,” *Empirical Software Engineering*, vol. 19, no. 6, pp. 1809–1855, 2014.

[107] F. Houdek and K. Pohl, “Analyzing requirements engineering processes: a case study,” in *Proceedings of 11th International Workshop on Database and Expert Systems Applications*, 2000, pp. 983–987.

[108] M. Weber and J. Weisbrod, “Requirements engineering in automotive development-experiences and challenges,” in *IEEE Joint International Conference on Requirements Engineering (RE ’02)*, 2002, pp. 331–340.

[109] N. Heumesser and F. Houdek, “Experiences in managing an automotive requirements engineering process,” in *12th IEEE International Requirements Engineering Conference (RE ’04)*, 2004, pp. 322–327.

[110] E. Sikora, B. Tenbergen, and K. Pohl, “Industry needs and research directions in requirements engineering for embedded systems,” *Requirements Engineering*, vol. 17, no. 1, pp. 57–78, 2012.

[111] E. Bjarnason, “Distances between requirements engineering and later software development activities: A systematic map,” in *Requirements Engineering: Foundation for Software Quality*, ser. Lecture Notes in Computer Science, J. Doerr and A. L. Opdahl, Eds., 2013, vol. 7830, pp. 292–307.

[112] M. Weinstein, “TAMS Analyzer,” last accessed Feb. 2015. [Online]. Available: http://tamsys.sourceforge.net/

[113] J. W. Creswell, *Research design: qualitative, quantitative, and mixed methods approaches*, 2nd ed. SAGE Publications, 2003, ch. 10.

[114] Scaled Agile Inc., “Scaled agile framework,” last accessed Feb. 2015. [Online]. Available: http://www.scaledagileframework.com/

[115] Atlassian, “JIRA - Issue & Project Tracking Software,” last accessed Mar. 2015. [Online]. Available: https://www.atlassian.com/software/jira

[116] S. Fricker, T. Gorschek, C. Byman, and A. Schmidle, “Handshaking with implementation proposals: Negotiating requirements understanding,” *Software, IEEE*, vol. 27, no. 2, pp. 72–80, 2010.

[117] T. Dingsøyr and N. B. Moe, “Research challenges in large-scale agile software development,” *SIGSOFT Software Engineering Notes*, vol. 38, no. 5, pp. 38–39, 2013.

[118] B. Fitzgerald, K.-J. Stol, R. O’Sullivan, and D. O’Brien, “Scaling agile methods to regulated environments: An industry case study,” in *35th International Conference on Software Engineering (ICSE ’13)*, 2013, pp. 863–872.

[119] E. Gottesdiener, “Requirements by collaboration: getting it right the first time,” *IEEE Software*, vol. 20, no. 2, pp. 52–55, 2003.

[120] R. Kylmäkoski, “Efficient authoring of software documentation using RaPiD7,” in *25th International Conference on Software Engineering (ICSE ’03)*, 2003, pp. 255–261.

[121] V. Sinha, B. Sengupta, and S. Chandra, “Enabling collaboration in distributed requirements management,” *IEEE Software*, vol. 23, no. 5, pp. 52–61, 2006.

[122] T. Kahkonen, “Agile methods for large organizations - building communities of practice,” in *Agile Development Conference 2004*, 2004, pp. 2–10.

[123] V. R. Basili, G. Caldiera, and H. D. Rombach, “Experience factory,” *Encyclopedia of software engineering*, vol. 1, pp. 469–476, 1994.

[124] A. Averbakh, E. Knauss, S. Kiesling, and K. Schneider, "Dedicated support for experience sharing in distributed software projects," in *26th International Conference on Software Engineering and Knowledge Engineering (SEKE'14)*, 2014.

[125] A. Vandevelde and R. Van Dierdonck, "Managing the design-manufacturing interface," *International Journal of Operations and Production Management*, vol. 23, no. 11-12, pp. 1326–1348, 2003.

[126] E. Bjarnason, K. Smolander, E. Engström, and P. Runeson, "Alignment practices affect distances in software development: A theory and a model," in *3rd SEMAT Workshop on General Theories of Software Engineering (GTSE '14)*, 2014, pp. 21–31.

[127] D. E. L. Maryam Alavi, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS Quarterly*, vol. 25, no. 1, pp. 107–136, 2001.

[128] N.-I. Boer, H. Berends, and P. van Baalen, "Relational models for knowledge sharing behavior," *European Management Journal*, vol. 29, no. 2, pp. 85–97, 2011.

[129] B. Hanisch, F. Lindner, A. Mueller, and A. Wald, "Knowledge management in project environments," *Journal of Knowledge Management*, vol. 13, no. 4, pp. 148–160, 2009.

[130] M. Bresnen, L. Edelman, S. Newell, H. Scarbrough, and J. Swan, "Social practices and the management of knowledge in project environments," *International Journal of Project Management*, vol. 21, no. 3, pp. 157–166, 2003.

[131] F. Lindner and A. Wald, "Success factors of knowledge management in temporary organizations," *International Journal of Project Management*, vol. 29, no. 7, pp. 877–888, 2011.

[132] F. O. Bjørnson and T. Dingsøyr, "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used," *Information and Software Technology*, vol. 50, no. 11, pp. 1055–1068, 2008.

[133] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *Journal of Systems and Software*, vol. 101, pp. 193–220, 2015.

[134] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice," *Software & Systems Modeling*, vol. 17, no. 1, pp. 91–113, 2018.

[135] G. Loniewski, E. Insfran, and S. Abrahão, "A systematic review of the use of requirements engineering techniques in model-driven development," in *ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MODELS '10)*, 2010, pp. 213–227.

[136] 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.

[137] A. Haghighatkhah, M. Oivo, A. Banijamali, and P. Kuvaja, "Improving the state of automotive software engineering," *IEEE Software*, vol. 34, no. 5, pp. 82–86, 2017.

[138] A. Vogelsang, S. Eder, G. Hackenberg, M. Junker, and S. Teufl, "Supporting concurrent development of requirements and architecture: A model-based approach," in *2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD '14)*, 2014, pp. 587–595.

[139] B. Berenbach, F. Schneider, and H. Naughton, "The use of a requirements modeling language for industrial applications," in *20th IEEE International Requirements Engineering Conference (RE '12)*, 2012, pp. 285–290.

[140] B. Penzenstadler, G. Haller, T. Schlosser, and G. Frenzel, "Soft Skills REquired: a practical approach for empowering soft skills in the engineering world," in *Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills (CIRCUS '09)*, 2009, pp. 31–36.

[141] E. van Akkeren, L. Baumann, J. J. Cannegieter, C. Hood, P. Hruschka, M. Lampe, E. Leutbecher, H. van Loenhoud, P. de Roo, S. Staal *et al.*, "Handbook of requirements modeling according to the IREB standard," 2016.

[142] M. Autili, L. Grunske, M. Lumpe, P. Pelliccione, and A. Tang, "Aligning qualitative, real-time, and probabilistic property specification patterns using a structured english grammar," *IEEE Transactions on Software Engineering*, vol. 41, no. 7, pp. 620–638, 2015.

[143] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Patterns in property specifications for finite-state verification," in *1999 International Conference on Software Engineering (ICSE '99)*, 1999, pp. 411–420.

[144] V. Gervasi and D. Zowghi, "Reasoning about inconsistencies in natural language requirements," *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 3, pp. 277–330, 2005.

[145] D. de Almeida Ferreira and A. R. da Silva, "A controlled natural language approach for integrating requirements and model-driven engineering," in *4th International Conference on Software Engineering Advances*, 2009, pp. 518–523.

[146] N. Marko, A. Leitner, B. Herbst, and A. Wallner, "Combining xtext and oslc for integrated model-based requirements engineering," in *41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA '15)*, 2015, pp. 143–150.

[147] D. Méndez Fernández, B. Penzenstadler, M. Kuhrmann, and M. Broy, "A meta model for artefact-orientation: fundamentals and lessons learned in requirements engineering," in *ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MODELS '10)*, 2010, pp. 183–197.

[148] D. Méndez Fernández, K. Lochmann, B. Penzenstadler, and S. Wagner, "A case study on the application of an artefact-based requirements engineering approach," in *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE '11)*, 2011, pp. 104–113.

[149] D. Méndez Fernández and R. Wieringa, "Improving requirements engineering by artefact orientation," in *14th International Conference on Product Focused Software Process Improvement (PROFES '13)*, 2013, pp. 108–122.

[150] B. Nuseibeh, "Weaving together requirements and architectures," *Computer*, vol. 34, no. 3, pp. 115–119, 2001.

[151] E. Hebisch, M. Book, and V. Gruhn, "Scenario-based architecting with architecture trace diagrams," in *5th International Workshop on the Twin Peaks of Requirements and Architecture (TwinPeaks '15)*, 2015, pp. 16–19.

[152] N. Rungta, O. Tkachuk, S. Person, J. Biatek, M. W. Whalen, J. Castle, and K. Gundy-Burlet, "Helping system engineers bridge the peaks," in *4th International Workshop on Twin Peaks of Requirements and Architecture (TwinPeaks '14)*, 2014, pp. 9–13.

[153] B. Paech, A. H. Dutoit, D. Kerkow, and A. Von Knethen, "Functional requirements, non-functional requirements, and architecture should not be separated - a position paper," in *8th International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ '02)*, 2002.

[154] P. Avgeriou, J. Grundy, J. G. Hall, P. Lago, and I. Mistrík, *Relating software requirements and architectures.* Springer Science & Business Media, 2011.

[155] U. Eliasson, R. Heldal, E. Knauss, and P. Pelliccione, "The need of complementing plan-driven requirements engineering with emerging communication: Experiences from volvo car group," in *IEEE 23rd International Requirements Engineering Conference (RE '15)*, 2015, pp. 372–381.

[156] M. Glinz, "Problems and deficiencies of uml as a requirements specification language," in *10th International Workshop on Software Specification and Design*, 2000.

[157] P. Mayring, "Qualitative content analysis–research instrument or mode of interpretation," *The role of the researcher in qualitative psychology*, vol. 2, pp. 139–148, 2002.

[158] R. K. Yin, *Case study: design and methods*, 4th ed., ser. Applied social research methods series; 5. Los Angeles, CA: Sage, 2009.

[159] D. North, “Introducing behaviour driven development,” *Better Software Magazine*, 2006. [Online]. Available: https://dannorth.net/introducing-bdd/

[160] H. Hofmann and F. Lehner, “Requirements engineering as a success factor in software projects,” *Software, IEEE*, vol. 18, no. 4, pp. 58–66, 2001.

[161] B. Graaf, M. Lormans, and H. Toetenel, “Embedded software engineering: the state of the practice,” *IEEE Software*, vol. 20, no. 6, pp. 61–69, 2003.

[162] C. J. Neill and P. A. Laplante, “Requirements engineering: the state of the practice,” *IEEE Software*, vol. 20, no. 6, pp. 40–45, 2003.

[163] D. Harel and S. Maoz, “Assert and negate revisited: Modal semantics for UML sequence diagrams,” *Software and Systems Modeling*, vol. 7, no. 2, pp. 237–252, 2008.

[164] J. Greenyer, C. Brenner, and V. Panzica La Manna, “The scenariotools play-out of modal sequence diagram specifications with environment assumptions,” in *12th International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT ’13)*, 2013.

[165] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, and W. Yi, *UPPAAL-a tool suite for automatic verification of real-time systems.* Springer, 1996.

[166] B. K. Aichernig, F. Lorber, and D. Nickovic, “Time for mutants - model-based mutation testing with timed automata,” in *7th International Conference on Tests and Proofs (TAP ’13)*, 2013, pp. 20–38.

[167] B. K. Aichernig, H. Brandl, E. Jöbstl, W. Krenn, R. Schlick, and S. Tiran, “Killing strategies for model-based mutation testing,” *Software Testing, Verification and Reliability*, vol. 25, no. 8, pp. 716–748, 2015.

[168] Y. Jia and M. Harman, “An analysis and survey of the development of mutation testing,” *IEEE Transactions on Software Engineering*, vol. 37, no. 5, pp. 649–678, 2011.

[169] B. K. Aichernig, H. Brandl, E. Jöbstl, and W. Krenn, “Uml in action: A two-layered interpretation for testing,” *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 1, pp. 1–8, 2011.

[170] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, and P. Mäder, *Traceability Fundamentals.* Springer London, 2012, pp. 3–22.

[171] “Road vehicles – Functional safety – Part 2: Management of functional safety,” *ISO 26262-2:2011*, pp. 1–26, 11 2011.

[172] B. Selic, G. Gullekson, J. McGee, and I. Engelberg, “Room: An object-oriented methodology for developing real-time systems,” in *5th International Workshop on Computer-Aided Software Engineering*, 1992, pp. 230–240.

[173] C. Heitmeyer, A. Bull, C. Gasarch, and B. Labaw, "Scr: a toolset for specifying and analyzing requirements," in *10th Annual Conference on Computer Assurance (COMPASS '95)*, 1995, pp. 109–122.

[174] P. Zave, "An operational approach to requirements specification for embedded systems," *IEEE Transactions on Software Engineering*, vol. SE-8, no. 3, pp. 250–269, 1982.

[175] K. Havelund, A. Skou, K. G. Larsen, and K. Lund, "Formal modeling and analysis of an audio/video protocol: an industrial case study using UPPAAL," in *18th IEEE Real-Time Systems Symposium*, 1997, pp. 2–13.

[176] H. Jensen, K. Larsen, and A. Skou, "Modelling and Analysis of a Collision Avoidance Protocol using SPIN and UPPAAL," *BRICS Report Series*, vol. 3, no. 24, 1996.

[177] K. Havelund, K. G. Larsen, and A. Skou, "Formal verification of a power controller using the real-time model checker uppaal," in *Formal Methods for Real-Time and Probabilistic Systems: 5th International AMAST Workshop*, 1999, pp. 277–298.

[178] K. G. Larsen, M. Mikucionis, B. Nielsen, and A. Skou, "Testing real-time embedded software using uppaal-tron: An industrial case study," in *5th ACM International Conference on Embedded Software*, 2005.

[179] L. Khatib, N. Muscettola, and K. Havelund, "Verification of plan models using UPPAAL," in *International Workshop on Formal Approaches to Agent-Based Systems (FAABS '00)*, 2000, pp. 114–122.

[180] B. P. Douglass, *Doing hard time: developing real-time systems with UML, objects, frameworks, and patterns.* Addison-Wesley Professional, 1999, vol. 1.

[181] F. Bordeleau, "A systematic and traceable progression from scenario models to communicating hierarchical state machines," Ph.D. dissertation, 1999, (Canadian dissertation).

[182] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Property specification patterns for finite-state verification," in *2nd Workshop on Formal Methods in Software Practice*, 1998, pp. 7–15.

[183] B. Nielsen and A. Skou, "Automated test generation from timed automata," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '01)*, 2001, pp. 343–357.

[184] M. Mikucionis, B. Nielsen, and K. G. Larsen, "Real-time system testing on-the-fly," in *15th Nordic Workshop on Programming Theory*, 2003, pp. 36–38.

[185] A. Hessel and P. Pettersson, "Cover - a test-case generation tool for timed systems," in *Testing of Software and Communicating Systems: Work-in-Progress and Position Papers, Tool Demonstrations, and Tutorial Abstracts of TestCom/FATES*, 2007, pp. 31–34.

[186] K. G. Larsen, F. Lorber, B. Nielsen, and U. M. Nyman, “Mutation-based test-case generation with ecdar,” in *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW ’17)*, 2017, pp. 319–328.

[187] F. Belli, M. Beyazit, T. Takagi, and Z. Furukawa, “Model-based mutation testing using pushdown automata,” *IEICE Transactions*, vol. 95-D, no. 9, pp. 2211–2218, 2012.

[188] D. E. Perry, N. A. Staudenmayer, and L. G. Votta, “People, organizations, and process improvement,” *IEEE Software*, vol. 11, no. 4, pp. 36–45, 1994.

[189] R. E. Kraut and L. A. Streeter, “Coordination in software development,” *Commun. ACM*, vol. 38, no. 3, pp. 69–81, 1995.

[190] J. D. Herbsleb and R. E. Grinter, “Architectures, coordination, and distance: Conway’s law and beyond,” *IEEE Software*, vol. 16, no. 5, pp. 63–70, 1999.

[191] M. Niazi, S. Mahmood, M. Alshayeb, M. R. Riaz, K. Faisal, N. Cerpa, S. U. Khan, and I. Richardson, “Challenges of project management in global software development: A client-vendor analysis,” *Information and Software Technology*, vol. 80, pp. 1–19, 2016.

[192] J. D. Herbsleb, “Global software engineering: The future of socio-technical coordination,” in *Future of Software Engineering (FOSE ’07)*, 2007, pp. 188–198.

[193] J. Srivastava, “Data mining for social network analysis,” in *IEEE International Conference on Intelligence and Security Informatics (ISI ’08)*, 2008, pp. 33–34.

[194] R. Cross, S. Borgatti, and A. Parker, “Making invisible work visible: Using social network analysis to support strategic collaboration,” *California Management Review*, vol. 44, no. 2, pp. 25–46, 2002.

[195] C. Y. Lin, L. Wu, Z. Wen, H. Tong, V. Griffiths-Fisher, L. Shi, and D. Lubensky, “Social network analysis in enterprise,” *Proc. IEEE*, vol. 100, no. 9, pp. 2759–2776, 2012.

[196] D. Damian, I. Kwan, and S. Marczak, “Requirements-driven collaboration: Leveraging the invisible relationships between requirements and people,” *Collaborative Software Eng.,J.Whitehead, I. Mistrik, J. Grundy, and A. van der Hoek, eds.*, pp. 57–76, 2010.

[197] T. Wolf, A. Schröter, D. Damian, L. Panjer, and T. H. Nguyen, “Mining task-based social networks to explore collaboration in software teams,” *IEEE Software*, vol. 26, no. 1, pp. 58–66, jan 2009.

[198] A. Begel, Y. P. Khoo, and T. Zimmermann, “Codebook: Discovering and exploiting relationships in software repositories,” in *32nd International Conference on Software Engineering (ICSE ’10)*, 2010, pp. 125–134.

[199] A. Mockus and J. D. Herbsleb, “Expertise browser: a quantitative approach to identifying expertise,” in *24th international conference on software engineering (ICSE ’02)*, 2002, pp. 503–512.

[200] AUTOSAR, “AUTOSAR,” last accessed Aug. 2015. [Online]. Available: http://www.autosar.org/

[201] A. Begel, N. Nagappan, C. Poile, and L. Layman, “Coordination in large-scale software teams,” in *Workshop on Cooperative and Human Aspects on Software Engineering (CHASE ’09)*. IEEE Computer Society, 2009, pp. 1–7.

[202] E. Otte and R. Rousseau, “Social network analysis: a powerful strategy, also for the information sciences,” *Journal of Information Science*, vol. 28, no. 6, pp. 441–453, 2002.

[203] S. L. Lim, D. Quercia, and A. Finkelstein, “Stakesource: Harnessing the power of crowdsourcing and social networks in stakeholder analysis,” in *32nd International Conference on Software Engineering (ICSE ’10)*, 2010, pp. 239–242.

[204] S. L. Lim, D. Damian, and A. Finkelstein, “Stakesource2.0: Using social networks of stakeholders to identify and prioritise requirements,” in *33rd International Conference on Software Engineering (ICSE ’11)*, 2011, pp. 1022–1024.

[205] A. Guzzi and A. Begel, “Facilitating communication between engineers with CARES,” in *34th International Conference on Software Engineering (ICSE ’12)*, 2012, pp. 1367–1370.

[206] A. Guzzi, A. Begel, J. K. Miller, and K. Nareddy, “Facilitating enterprise software developer communication with CARES,” in *28th IEEE International Conference on Software Maintenance (ICSM ’12)*, 2012, pp. 527–536.

[207] V. Vaishnavi and B. Kuechler, “Design research in information systems.” *Association for Information Systems*, 2004.

[208] A. Fink, *The survey handbook.* Sage, 2003, vol. 1.

[209] Volvo Group, “Volvo group,” http://www.volvogroup.com, 2016, last accessed Apr. 2016.

[210] SYSTEMITE AB, “Systemweaver [computer program],” http://www.systemweaver.se/, 2014, last accessed Apr. 2016.

[211] ——, “swexplorer reference manual,” 2015.

[212] A. Zaczek and D. Schmitt, “Graph#,” https://graphsharp.codeplex.com/, last accessed Apr. 2016.

[213] A. Cooper, R. Reimann, D. Cronin, and C. Noessel, *About Face: The Essentials of Interaction Design.* Wiley, 2014.

[214] C. Ware, *Information Visualization: Perception for Design*, ser. Information Visualization: Perception for Design. Morgan Kaufmann, 2013.

[215] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, “A systematic literature review on agile requirements engineering practices and challenges,” *Computers in human behavior*, vol. 51B, pp. 915–929, 2015.

[216] 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.

[217] OSLC, “Open Services for Lifecycle Collaboration,” https://open-services.net/, last accessed Jun. 2017.

[218] “Ieee standard glossary of software engineering terminology,” *IEEE Std 610.12-1990*, pp. 1–84, Dec 1990.

[219] D. Howe, *Free Online Dictionary of Computing*, http://foldoc.org/.

[220] “IEC 62531:2012(E) (IEEE Std 1850-2010): Standard for Property Specification Language (PSL),” *IEC 62531:2012(E) (IEEE Std 1850-2010)*, pp. 1 – 184, June 2012.