Softw Syst Model manuscript No.
(will be inserted by the editor)

Model-Based Engineering in the Embedded Systems Domain

An Industrial Survey on the State-of-Practice

Grischa Liebel[1], **Nadja Marko**[2], **Matthias Tichy**[3], **Andrea Leitner**[2], **Jörgen Hansson**[4]

[1] Software Engineering Division, Chalmers | University of Gothenburg, Sweden
e-mail: `grischa@chalmers.se|matthias.tichy@cse.gu.se`
[2] Virtual Vehicle Research Center, Graz, Austria
e-mail: `nadja.marko@v2c2.at|andrea.leitner@v2c2.at`
[3] Institute of Software Engineering, University of Ulm, Germany
e-mail: `matthias.tichy@uni-ulm.de`
[4] School of Informatics, University of Skövde, Sweden
e-mail: `jorgen.hansson@his.se`

Received: date / Accepted: date

Abstract Model-Based Engineering (MBE) aims at increasing the effectiveness of engineering by using models as important artifacts in the development process. While empirical studies on the use and the effects of MBE in industry exist, only few of them target the embedded systems domain. We contribute to the body of knowledge with an empirical study on the use and the assessment of MBE in that particular domain. The goal of this study is to assess the current State-of-Practice and the challenges the embedded systems domain is facing due to shortcomings with MBE. We collected quantitative data from 113 subjects, mostly professionals working with MBE, using an online survey. The collected data spans different aspects of MBE, such as the used modeling languages, tools, notations, effects of MBE introduction, or shortcomings of MBE. Our main findings are that MBE is used by a majority of all participants in the embedded systems domain, mainly for simulation, code generation, and documentation. Reported positive effects of MBE are higher quality and improved reusability. Main shortcomings are interoperability difficulties between MBE tools, high training effort for developers and usability issues. Our study offers valuable insights into the current industrial practice and can guide future research in the fields of systems modeling and embedded systems.

Key words Model-Based Engineering Model-Driven Engineering Embedded Systems Industry Modeling Empirical Study State-of-Practice

1 Introduction

Embedded systems are systems that are “integral components of larger systems”, which are used to “control and/or directly monitor that system using special hardware devices” [2]. This includes systems from large industries such as automotive, avionics, health care, and railway. Building embedded systems is considered a complex process due to their scale and distribution [21], the interplay between multiple disciplines [11] and due to high availability and safety demands [10]. Additionally, complexity in many embedded systems is increasing rapidly due to the increased amount of functionality taken over by software [10]. For instance, in year 2002, the Volvo XC90 automobile contained 38 Electronic Control Units (ECUs) [28]. The 2015 model already contains 108 ECUs [28], almost three times as many as in the 2002 model.

Given the high complexity in embedded systems engineering, modeling is used to tackle this complexity and improve the development of embedded systems. For example, the first version of Matlab/Simulink has been released exactly 30 years ago and is by now one of the standard development tools in the automotive domain. Additionally, in order to support embedded systems engineering, standards such as MARTE [29] extend UML [30], the de-facto standard modeling language in the field of software engineering.

In particular, Model-Based Engineering (MBE) and Model-Driven Engineering (MDE) are engineering approaches that employ models to handle complexity by means of abstraction from the problem [14]. We use the definitions of Brambilla et al. [7], in which MBE comprises approaches were models play an important role but not necessarily the primary role. Model-Driven Engineering (MDE) has a more narrow scope and is an approach in which models are used as the primary artifacts throughout the entire engineering process [7]. Throughout this paper, we use the term MBE as a comprehensive term that includes MDE and other, more restrictive variants such as Model-Driven Architecture or Model-Driven Development.

Despite the long history of MBE and the fact that MBE aims to increase the effectiveness and efficiency of software development, empirical evaluation of the use of MBE in industry is scarce [26]. Therefore, it is unclear whether MBE methods are really accepted and if any potential advantages are recognized in industry. Many MBE methods and theoretical advantages are discussed in research and our personal experience, from industry and research projects in collaboration with industry, indicates that MBE is used widely. Nevertheless, there is no empirical data to factually support this. Similarly, there is only limited knowledge regarding advantages and shortcomings arising from the use of MBE.

The few existing empirical studies in this field suggest that MBE can have positive effects such as reduction of defects and productivity improvements [6,26], or increased understandability [20]. Some challenges have been reported, such as insufficient tool support [6,26,27], need for additional training [20], or the use of MBE with legacy software [26,20]. However, existing studies are not explicitly targeted at the embedded systems domain [6, 26,27,18,20,34], but rather target only UML [16,5,9], limit themselves to the Brazilian embedded industry [4], or collect only qualitative data from the automotive domain [22]. Hence, we decided to contribute to the body of knowledge with a survey on the use of MBE in the embedded systems domain. We are interested in how industry actually applies MBE, i.e., which methods, languages, and tools are used, and which challenges arise with the application of MBE. Further, we think that the identification of differences between subgroups of users, such as users from different domains or company sizes, could yield a deeper understanding about challenges that are possibly solved by several subgroups.

In short, the main goal of the survey is to get an overview about the State-of-Practice (SoP) and challenges the industry is faced with in order to understand industrial needs. More specifically, the study aims to answer the following research questions:

- **RQ1:** What is the current state of practice and the assessment of MBE in the embedded systems domain?
- **RQ2:** How does the use and the assessment of MBE differ among different demographic subgroups in the embedded systems domain?

RQ1 aims to capture the SoP of MBE in the embedded systems domain, which includes the used modeling environments, modeling languages, types of notations, purposes of models are used for, and how much activities concern MBE compared to non-MBE. Moreover, we are interested in understanding reasons and the effects, both positive and negative, when adopting and deploying MBE. With **RQ2**, we want to assess whether there are substantial differences in the SoP between different groups in the embedded systems domain, e.g., differences in the automotive domain and the avionics domain, or between new MBE users and highly experienced MBE users.

In order to answer these research questions, we developed a web survey consisting of 24 questions. The survey was distributed to partners taking part in at least one of five industrially-driven European research projects (between 22 and 100 project partners) as well as to personal industry contacts working with MBE. We received 121 finished surveys from which 113 were used for the data analysis.

In this paper, we present the results of the survey with focus a on the effects of MBE, shortcomings, and introduction needs. Moreover, we compare our results with related work based on 25 hypotheses extracted from related work and 8 hypotheses defined by us specifically for **RQ2**. Overall, the survey answers show that most survey participants think that the positive effects distinctly exceed the negative effects of MBE. Nevertheless, they mention also, e.g., that interoperability challenges between tools exist and that MBE causes high efforts to train the developers. More detailed results are discussed in Section 4.

The remainder of this paper is structured as follows. In the next section we discuss related MBE studies. Section 3 contains the research methodology. This includes the process of study design, data collection, and threats to validity. In Section 4, the key results of the survey are discussed. Finally, conclusions and future work are discussed in Section 5.

2 Related Work

Industrial evaluation of MBE in research is limited [20], but there have recently been a number of publications addressing the topic. With respect to the embedded systems domain, we are only aware of two reported studies presenting the SoP of MBE [4,22]. Other publications, such as [6,26,18] and [19], also include cases from the embedded systems domain, but do not explicitly address this domain as their target. We now turn our attention to introducing the different areas of related work in detail, starting with work directly focused on MBE in the embedded industry, followed by empirical studies on MBE in a wider context, and concluding with empirical studies on modeling in general.

2.1 Empirical Studies on MBE in the Embedded Industry

Agner et al. present the results of a survey on the use of UML and model-driven approaches in the Brazilian embedded software development industry [4]. The participants come from a variety of different subdomains, with industrial automation, information technology, telecommunications and electronic industry being the biggest groups. The findings show that 45% of the 209 participants use UML; a majority of these UML users are experienced developers working at medium-sized companies. Increases in productivity and improvements in quality, maintenance and portability as key advantages of model-driven practices are reported. According to the participants, the use of UML is mostly hindered by short lead times, lack of knowledge regarding UML and a limited number of employees with expert UML knowledge. Additionally, it is stated that models are mainly used for documentation with only little use of code generation or model-centric approaches in general. In contrast to Agner et al's work [4], we do not limit ourselves to a region but include a wide range of subjects from global companies based in Europe.

Kirstan and Zimmermann report a case study within the automotive domain [22]. Their interviewees report positive effects of MBE like an earlier detection of errors, a higher degree of automation and cost savings during the initial phases of development. However, they also state that large function models can become too complex and that interoperability between tools is difficult. The study is limited to qualitative data in the automotive domain, thus a sub-domain of embedded systems.

2.2 Empirical Studies on MBE in General

Baker et al. present experiences with MBE at Motorola over a 20-year time span [6]. They report positive effects such as a reduction in defects and an improvement in productivity. However, a number of challenges regarding MBE are named as well, including lack of common tools, i.e. tools that support different languages or subsets of languages, which makes it difficult to exchange models between development groups using different tools. Additionally, poor tool and generated code performance is reported, specifically that tools do not scale well to industrial-size models and that, thus, the code generated from the models can cause performance bottlenecks.

Mohagheghi and Dehlen published a literature review on the industrial application of MBE [26]. The findings suggest that MBE can lead to improvements in software quality and productivity. However, studies which report productivity losses are also identified. Costly integration of MBE tools, modeling complexity, and the use of MBE with legacy systems are reported as challenges. Additionally, the maturity of tool environments is stated to be unsatisfactory for a large-scale adoption of MBE. Generally, the authors conclude that there is too little evidence in order to draw any general conclusions from their results.

In a later publication by Mohagheghi et al., experiences from three companies in a European project “with the objective of developing techniques and tools for applying MDE” are reported [27]. According to the experiences at the studied companies, advantages of using MBE include the possibility to provide abstractions of complex systems, simulation and testing, and performance-related decision support. However, the authors also state that the development of reusable solutions using MBE requires additional effort and might decrease performance. Moreover, transformations required for tool integration can increase the complexity and the implementation effort according to the authors. Furthermore, the user-friendliness of MBE tools and means for managing models of complex systems is described as challenging.

Hutchinson et al. report industrial experiences from the adoption of MBE at a printer company, a car company, and a telecommunications company [18]. The authors conclude that a successful adoption of MBE seems to require, among others, an iterative and progressive approach, organizational commitment, and motivated users. Another study by the same authors, in which the interview data is complemented by survey data, points out that it is not sufficient to consider only the technical challenges and benefits of MBE, but that they should be placed in the wider organizational context [19]. Similarly, Whittle et al. point out that tooling in MBE is a problem, but organizational issues need to be considered as well [35]. These three publications do not primarily consider embedded systems engineering, and are mainly focused on organizational challenges of MBE.

A further assessment of MBE in industry, conducted by Hutchinson et al. [20], is based on more than 250 survey responses, 22 interviews, and observational studies from multiple domains. The authors report that significant additional training is needed for the use of MBE, but that MBE in turn can speed up the implementation of new requirements. Furthermore, the survey indicates that code generation is an important aspect of MBE productivity gains, but integrating the code into existing projects can be problematic. The majority of survey participants states that MBE does increase understandability. From their interviews, the authors conclude that people’s ability to think abstractly appears to have significant impact on their ability to model. Hence, this ability influences the success of MBE.

According to a survey of 113 software practitioners reported by Forward and Lethbridge, common problems with model-centric development approaches are, among others, inconsistency of models over time, model interchange between tools, and heavyweight modeling tools [13]. However, code-centric development approaches make it difficult to see the overall design and make it hard to understand the system behavior.

Torchiano et al. present findings from a survey on the SoP in model-driven approaches in the Italian software industry [34]. From the 155 subjects, 68% report to always or sometimes use models. The subjects who do not use models commonly state that modeling requires too much effort (50%) or is not useful enough (46%). Further findings are that models are used mainly in larger companies and that a majority (76%) of all the interviewees using models apply UML.

2.3 Empirical Studies on UML

Numerous empirical studies exist that target UML as an example of modeling languages. Hereby, studies that report effects of using or introducing UML are of particular interest for our study.

Grossman et al. report a survey of 131 UML users and state that UML is seen as “accurate, consistent, and flexible enough to use on development projects” by the participants [16]. However, they also report that they are lacking deep enough understanding in order to judge whether UML actually makes a real difference.

A number of improvements following the introduction of UML in a project at ABB are reported by Anda et al. [5]. Their data from 16 interviews with project managers and developers shows positive effects of introducing UML, such as improvements in traceability from requirements to code, improved communication within teams, or better documentation. Reported difficulties are choosing the right diagrams and the right level of detail.

While these studies are targeting UML, they could equally well applied to MBE when UML is used as a modeling language.

2.4 Summary

In conclusion, commonly reported problems related to use of MBE include insufficient tool support or tool chains, using MBE together with legacy systems, and the complexity of MBE and modeling in general. Positive outcomes are reported as well, such as productivity gains, defect reductions, and increased understandability are reported. However, there is a lack of empirical evidence and reported industry evaluations on the use of MBE within the embedded systems domain. Existing work is either not targeting the embedded systems domain in particular, is limited to the Brazilian market, lacks quantitative data, or targets modeling in general without a particular MBE focus.

3 Research Methodology

This section outlines the research methodology, consisting of the study design and threats to validity. The study design summarizes the procedure of survey development including survey design, an outline of data collection, and data analysis as well as the hypothesis definitions for comparison with related work and analysis of differences in subgroups. The threats to validity section outlines threats which can influence the results of the survey.

3.1 Study Design

The study was designed by three researchers and three practitioners, each from two different companies as part of the European CRYSTAL project [8]. As, according to our knowledge, MBE is already widely implemented in the automotive sector and the distribution has progressed similarly in other industrial branches, we decided to perform a survey as they are suitable for collecting empirical data from large populations. Further, we chose to perform an online survey for data collection in order to minimize the effort for both, participants and researchers. Based on these decisions, we followed the process shown in Figure 1 which is based on [31] and [32]. Punter et al. picture in their work a survey design process with focus on performing online surveys [32]. This process consists of nearly the same activities our survey is based on, namely *Study definition*, which represents the definition of the goal and the research questions in our case, *Study design*, *Implementation* which also includes the validation in their work, *Execution* which represents the data collection, *Analysis* and *Packaging.* Additionally, we elicited and collected hypotheses, independent of the survey design, which are analyzed against the survey data. The results are summarized in Section 4. Additionally, the data is packaged within a technical report in [25].

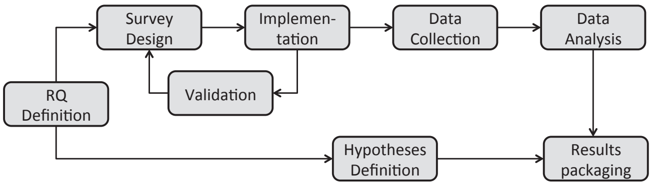

Fig. 1 Study Design activities

3.1.1 Survey Design and Implementation Based on the research questions we developed a questionnaire which consists of two main parts:

1. Context: In this part, information is gathered about the participants’ working environment and their per-

sonal experiences in order to be able to better interpret results and analyze data with regard to differences in demographic subgroups **(RQ2)**. We asked for company size, position in the value chain, domain, experience with MBE, product size, working tasks, and the attitude towards MBE, for this purpose.

2. MBE Practice: The second part of the questionnaire contains questions about the MBE State of Practice as well as an assessment of MBE **(RQ1)**. Here, we asked for used modeling languages, notation types, tools, introduction reasons, shortcomings, and effects of MBE.

We used an online survey[3] for implementation in order to keep administration costs low and facilitate distribution. After the implementation of the questionnaire in the online survey tool, the link to the survey was distributed to eleven colleagues in academia and industry who then validated the questionnaire and checked the understandability of the questions. Given their feedback and the time they needed to fill out the survey, the questionnaire was refined. The revised survey was reviewed a second time by one colleague not included in the pilot survey. The final survey questionnaire consists of a short introduction text describing the survey, and 24 questions of whereof 13 questions aim at gathering demographic data (context) and the remaining eleven questions address **RQ1** (MBE Practice). Both parts of the questionnaire were considered together for answering **RQ2**. Nearly all questions provided single-choice or multiple-choice answers which is most convenient for automatic statistical analysis. Where applicable, free-text areas for additional input were provided in order to avoid limitations in answers. Necessary definitions of the terms used in the questionnaire are listed in Appendix A.

3.1.2 Hypotheses Definition We derived a list of 25 hypotheses from the related work discussed in Section 2 (see Table 1) in order to guide the data analysis for **RQ1**. These were then evaluated based on the collected data. The hypotheses were grouped into three different groups depending on whether they address advantages of MBE ($H_{adv}.X$), challenges of/with MBE ($H_{chall}.X$), or the adoption of MBE ($H_{adopt}.X$). Where similar statements are found in multiple sources (see source column of Table 1), the presented hypotheses are summaries of the actual statements in the related publications. For instance, Hypothesis $H_{chall}.2$ describes tool quality in general, while Baker et al. talk about poor tool performance [6], Mohagheghi and Dehlen report lack of maturity of third-party tool environments [26], Mohagheghi et al. report challenges with the user-friendliness of tools [27], and Forward et al. report that heavyweight modeling tools are problematic [13]. While we lose the exact statements for these summarized hypotheses, we argue that this summary is helpful for getting an overview of the findings in the area of MBE. Hypothesis $H_{adopt}.5$ is not derived from related work, but was raised as a question during the discussion of our paper at MODELS 2014 [24]. Multiple people in the audience expressed that the large number of subjects using code generation in our survey could, in fact, be related to the low abstraction level commonly used in Matlab/Simulink/Stateflow[1] models. We therefore chose to add this hypothesis to our list and discuss it in this extended version. We do not claim that this list of hypotheses is complete. However, we believe that it can guide future research in this area.

Additionally, we derived a list of eight hypotheses in order to answer **RQ2**. We derived these hypotheses after designing our questionnaire from our own view on MBE. That is, we elicited the hypotheses based on the demographic subgroups which we were able to distinguish in our survey. The null hypotheses are in all cases that there are no significant differences between the subgroups. The alternative hypotheses, that there are significant differences between the subgroups, are listed in Table 2.

In-house tooling is specifically tailored towards the needs of the company which owns the tooling. Therefore, we would expect that users of in-house tooling are more positive towards the effects of MBE (**H2.1**). Hypothesis **H2.2** reflects that, commonly, supporters of a paradigm or a methodology perceive its advantages much more positively than subjects who do not support it. Therefore, we tested this hypothesis for the case of MBE supporters, and MBE opponents or neutral participants. Similarly, it could be expected that participants who still use MBE also see more positive effects of MBE than participants who stopped using MBE (**H2.3**). MBE can offer a multitude of benefits, such as automatic correctness checks or code generation. We would expect that participants who only use their models for documentation or information purposes and do not exploit the full potential will report less positive than negative effects. This is reflected by Hypothesis **H2.4**. Tooling in MBE is often reported to be insufficient. We would expect that usability issues with tools also influence other aspects such as productivity or quality negatively. Therefore, we investigate whether subjects who see many usability issues with MBE tools also report more negative effects than other subjects (**H2.5**). Apart from being insufficient, the complexity of MBE tools is often brought up as a challenge. This suggests that the learning curve is very high in the beginning, indicating that it should be easier to use MBE tools once you are experienced in MBE. Hence, we derived the hypothesis that experienced MBE users report less problems than MBE novices (**H2.6**). Hypothesis **H2.7** targets the complexity of large organizations. As their tool landscape often has a bigger variety, possibly including sub-contractors, tool integration

---

[3] through www.soscisurvey.de

[1] In the following, we state only Matlab instead of Matlab/Simulink/Stateflow

Table 1 Hypotheses from related work

Hypothesis	Description	Source
$H_{adv}.1$	MBE leads to a reduction of defects/improvements in quality.	[6,26,4]
$H_{adv}.2$	MBE leads to improvements in productivity.	[6,26,4]
$H_{adv}.3$	MBE increases understandability.	[20], partly [27]
$H_{adv}.4$	Advantages of MBE are simulation and testing, and performance-related decision support.	[27]
$H_{adv}.5$	MBE leads to an earlier detection of errors.	[22]
$H_{adv}.6$	MBE can speed up the implementation of new requirements.	[20]
$H_{adv}.7$	Code generation is an important aspect of MBE productivity gains.	[20]
$H_{adv}.8$	MBE leads to a higher degree of automation.	[22]
$H_{chall}.1$	Using MBE with legacy systems is challenging.	[26,20]
$H_{chall}.2$	Current MBE tools are insufficient.	[6,26,27,13, 22]
$H_{chall}.3$	Significant additional training is needed for using MBE.	[20,4]
$H_{chall}.4$	Managing models of complex systems is challenging.	[27,22]
$H_{chall}.5$	Tool integration is challenging.	[13,22,27]
$H_{chall}.6$	Code generated from models has poor performance.	[6]
$H_{chall}.7$	MBE lacks scalability.	[6]
$H_{chall}.8$	The complexity of modeling is challenging.	[26]
$H_{chall}.9$	Companies which consider software development their main business seem to find the adoption of MBE more challenging than other companies.	[20]
$H_{chall}.10$	Modeling requires too much effort.	[34]
$H_{chall}.11$	Handling the consistency of models over time is challenging.	[13]
$H_{chall}.12$	Modeling is not useful enough.	[34]
$H_{adopt}.1$	UML is the preferred modeling language employed in MBE.	[34,4]
$H_{adopt}.2$	Models are used mainly in larger companies.	[34]
$H_{adopt}.3$	UML is mostly used by experienced developers working at medium-sized companies.	[4]
$H_{adopt}.4$	There is little use of code generation or model-centric approaches.	[4]
$H_{adopt}.5$	Code generation is mainly used by subjects using Matlab/Simulink/Stateflow.	Conference discussion

is imposing a larger overhead and will most likely lead to more challenges. Finally, Hypothesis **H2.8** states that supporters of MBE also use more MBE tools in comparison to subjects who are opposed to or neutral towards MBE. As subjects using MBE tools during most of their working time are usually more familiar with their usage, we expect that these subjects do not share the same frustration as subjects using MBE tools only sporadically.

3.1.3 Data Collection The theoretical target population of the survey are people involved with systems engineering from the embedded systems domain, e.g. software architects, software developers, project managers, system engineers. We applied cluster-based sampling and distributed the survey via email to partners taking part in the Artemis projects Crystal (70 partners), VeTeSS (22 partners), MBAT (38 partners), nSafeCer (29 partners), and EMC^2 (100 partners), as well as to personal contacts of which most are professionals working with MBE (convenience sampling). Further, we encouraged recipients to distribute the survey to colleagues or partners (snowball sampling). The distribution to various people aimed at obtaining a large, heterogeneous sample. However, the introduction emails and introduction text in the questionnaire clearly stated that the survey was targeting people working with embedded systems.

The final version of the survey was published on 18th October 2013 for a time period of six weeks. During that time the number of answers has been checked periodically and intermediate results have been downloaded in order to discover possible errors or problems at an early stage. Finally, 121, out of 196 started surveys, were completed corresponding to a completion rate of 61.73%. In our survey, completed means that the participants navigated until the last page and pressed the submit button. We did not pose any compulsory questions.

3.1.4 Data Analysis The survey data was automatically coded and enhanced with additional quality data by the survey tool, such as completed answers and time to fill out the survey. We cleaned the remaining 121 surveys based on degradation points computed from missing answers and the time to fill out each survey page. Since we did not use compulsory questions, it could happen that subjects lost their interest but still navigated through the entire survey until the end or simply looked at the survey without filling in data. Therefore, we argue that this data cleaning process is necessary in order to

Table 2 Hypotheses defined for RQ2

Hypothesis	**Description**
H2.1	Users of in-house tools report more positive and less negative effects of MBE than users who do not use in-house tools.
H2.2	Supporters of MBE report more positive effects than subjects opposed to or neutral towards MBE.
H2.3	Subjects who are still using MBE report more positive and less negative effects than subjects who stopped using MBE.
H2.4	Subjects who only use models for means of information/documentation report less positive than negative effects.
H2.5	Subjects who do not see many usability issues with MBE tools report fewer negative effects.
H2.6	Highly experienced users of MBE report less problems with MBE tools than users with less experience.
H2.7	Large companies have more tool integration problems than small or medium-sized enterprises.
H2.8	MBE promoters use more MBE tools in comparison to subjects neutral or opposed to MBE.

ensure data validity as discussed in [36]. In our previous paper [24], we excluded nine surveys based on a threshold of 200 degradation points proposed by the survey tool for a light data filtering. These degradation points are based on missing answers and time spent on answering the questions compared to the median of all filled out surveys. However, experimental data from Leiner suggests that using missing answers is not well-suited for identifying meaningless data [23]. Additionally, the degradation points are calculated from all survey pages in the used survey tool, including the start and end pages. In our case, these only contain text and no questions, which causes the degradation points to be skewed. Therefore, we use an adapted data cleaning process for the data analysis in this paper, based only on the time spent on the actual survey pages as proposed in [23]. As a cut-off point, we chose survey answers in which subjects were in average twice as quick as the median time on each survey page containing questions. This left us with 113 answered surveys for data analysis, of which 108 were also included in the data analysis in [24]. We also made adaptations to the demographic data in cases where free-text answers clearly corresponded to one of the given answering options. The complete data sample together with the questionnaire and the data analysis scripts is published at `http://grischaliebel.de/data/research/sosym_LMTLH.zip`.

3.2 Validity Threats

In the following, we discuss the four different aspects of validity as discussed in Wohlin et al. [36].

3.2.1 Construct Validity Construct validity reflects whether the studied measures are generalizeable to the concept underlying the study. We collected data from different sources in order to avoid mono-operation bias. Hypothesis guessing, the participants guessing what the researchers are aiming for and answering accordingly, can not be ruled out completely. We tried, however, to formulate the questions in a neutral way and improved the questionnaire based on obtained feedback from the pilot study in order to address this threat. Finally, answers were treated completely anonymous in order to avoid biased answers due to evaluation apprehension.

3.2.2 Internal Validity Internal validity reflects whether all causal relations are studied or if unknown factors affect the results. The instrumentation quality was improved by using a two-round pilot study. The survey took approximately 15 minutes to fill out and was intended to be filled out once by every participant. This reduces the likelihood for learning effects and, hence, maturation effects. Additionally, the completion rate of 61.73% indicates that the majority of participants was interested in finishing the survey. Selection threats can not be ruled out as participants volunteered to fill out the survey. This voluntary participation and the fact that most of our contacts are through research projects concerned with MBE might skew the data being overly positive towards MBE. This should be taken into consideration when interpreting the results regarding the introduction effects of MBE. With respect to causal influence, we have to trust the subjects' domain knowledge. However, we can not rule out the risk that we or the subjects overlooked causal effects. For example, positive or negative effects, which subjects observed in conjunction with introducing MBE might, in fact, be attributed to other changes made together with the introduction.

3.2.3 External Validity External validity is concerned with the generalizeability of the findings. The CRYSTAL project and other projects, to which partners the survey was distributed, consist of partners from all major subdomains of the embedded systems domain. Additionally, demographic data was collected in order to confirm this aspect. Therefore, we are confident that we have reached subjects with a variety of different backgrounds representative for the embedded systems domain. Given that we distributed the survey mainly in European research projects, the generalizeability could be limited to a European level. However, many partners involved in the research projects operate on a global scale, have offices in many countries on multiple continents, or are owned by non-European organizations. Additionally, European suppliers often work with Original Equipment Manufacturers (OEMs) from outside Europe, or vice versa. Therefore, we believe that other factors, such as the domain or the position in the value chain, will in fact have a higher impact on the survey outcomes. To reduce this threat, we plan to replicate the study with companies outside of Europe.

3.2.4 Conclusion Validity Conclusion validity is concerned with the ability to draw correct conclusions from the studied measures. We involved three researchers and three practitioners with different background into the study design. Therefore, the survey was designed by multiple people with different aims and backgrounds, which should reduce the risk for “fishing” for results. A standard introduction e-mail was designed to be distributed with the link to the online survey. Hence, reliability of treatment implementation is given. We chose to provide no definition of MBE to the subjects, but instead targeted one of the questions at how they see MBE. While this introduces a certain bias that MBE could be understood in very different ways, we believe that this reflects the confusion around the different terms in both research and industry. Additionally, we discuss the outcome of the named question in our data analysis in order to show in which ways MBE was understood by the subjects. As the survey invitations were initially distributed through our professional networks, we aim to limit the heterogeneity in our subjects. Reliability of measures was increased through a survey pilot filled out by eleven people and then, after improvements, reviewed by one more researcher. However, we later encountered multiple questions with unclear wording, which could have lead to unreliability of our measures. These questions were therefore excluded in the data analysis. The detailed questionnaire is furthermore published in order to enable replications and an assessment of the validity of our study. Significance tests were only performed based on our hypotheses. That is, we performed only a fixed number of statistical tests and did not randomly search for significant results.

4 Results

In this section, we summarize the results of the survey. First, we illustrate demographic data about the subjects participating in the survey in order to get information about their company and experiences. Then, we address **RQ1** from different perspectives, such as effects and shortcomings of introducing and using MBE. Finally, we discuss **RQ2**. Both sections on **RQ1** and **RQ2** include a part dedicated to the evaluation of the hypotheses presented in Section 3.

4.1 Demographic Data

The first part of the survey contains context questions providing demographic data. We mainly asked for two kinds of background information: questions concerning the company and questions about the personal MBE experiences. With the company related questions, we wanted to get an idea of the work environment such as the domain, company size, or the company’s position in the value chain. Questions about the personal experiences such as daily working tasks, usage of MBE or whether the participant is a supporter for MBE or not, were asked in order to better understand answers and predispositions of the interviewees.

4.1.1 Company Context A bit more than the half of the 113 participants reported the company they work for, resulting in a list of 30 different companies. About three-fourths of all respondents (88) work in large companies with more than 250 employees, 13 persons are employed in small and medium sized enterprises (SME), and 11 work in academia or other research institutes. Hence, the main percentage of answers represent opinions of persons employed in large companies.

49 of the companies are first-tier suppliers, 41 are OEMs, 25 are second-tier suppliers, and 19 have other positions in the value chain such as research institutes, consultants or technology/software providers.

More than a half of the respondents (62) work in the automotive industry, 32 in avionics, 26 in health care, 16 in defense industry, 12 in rail industry, and 3 in telecommunications. 16 respondents work in multiple domains (i.e. domain independent) and 9 operate in other domains such as semiconductor or industrial automation industry. In order to understand how much experience the company has in using MBE, we asked participants when their companies started to use MBE. 35 say that their company introduced MBE 10 or more years ago, 55 state 1-10 years ago and 4 started in the last 12 months. 9 companies still do not apply MBE, the remaining 10 participants do not know the introduction time. Thus, most companies have experiences with MBE for quite some time. Though the introduction time of MBE gives a rough estimate of a company’s experience, we also wanted to learn the scale of target products that are developed using MBE. 74 companies use MBE for developing a commercial product, 49 companies use it for large-scale series production (more than 1000 pieces), 17 for medium-scale production, and 8 for small-scale production (less than 10 pieces). 24 use MBE for research demonstrations, 9 use it for non-commercial products, and 6 for other purposes such as teaching or developing methods and tools. Hence, generally survey participants work in companies with high experience levels coming from different domains which ensures that companies apply MBE and understand MBE challenges.

4.1.2 Personal Experiences In order to understand in which context the participants use MBE, we asked for their main working tasks. Here, multiple answers were possible, and are summarized as follows: 62 of the participants implement software, 57 are responsible for architecture definition, 58 work in testing, 56 work in design definition, 49 specify requirements, 40 are project managers, 25 are safety managers, 15 are quality managers, 14 are responsible for customer support, and 12 work in general management. 14 participants carry out other activities than the ones mentioned before, such as process improvement, consulting, or tool engineering. Hence, we cover a wide range of subjects working in different functions.

Many participants (48) have more than three years experience in using MBE. They can, thus, be considered to have vast experience. 40 persons state that they have moderate experience; 25 are new in the field of MBE.

Additionally, we wanted to know if survey participants currently use MBE or if they have stopped using it. 74 of the participants are still using MBE, 15 people used MBE the last month or up to one year ago, and 15 people used MBE the last time more than one year ago. Only 9 people state that they have never used MBE. Hence, 92% of the surveyed subjects have already applied MBE and have therefore some experience with it.

Finally, we asked participants for their attitude in order to get an understanding of how they perceive MBE. Results show that many persons have a positive attitude towards MBE. 87 people promote MBE, 24 have a neutral attitude towards it, and one is an opponent of MBE.

Based on the collected data we conclude that participants of this survey predominantly work in companies in which MBE is accepted and that they have good experiences with MBE.

4.2 RQ1: State of Practice and Assessment of MBE

The survey data offers valuable insights into the industrial practice of MBE. Based on the survey results, we provide in the following information about used methods and tools, purposes of models, effects of using it, as well as shortcomings of MBE. Additionally, we asked for reasons why MBE is applied or rather the motivation for the introduction.

4.2.1 Modeling Tools and Languages We asked participants about languages, notations, and tools they use for modeling, and which functional aspects of their system they describe using models in order to find out technical aspects that are applied in industry. For all questions, participants had to state if the method is used personally and if it is used somewhere in the department or division. The answers for the personal usage help to better understand the correlation between applied methods and the participants' assessment of MBE, whereas the department-level answers give information about the quantity of application of methods/tools.

The question about used tools shows that most survey participants use Matlab (50 personal/83 department) or Eclipse-based (35 personal/47 department) tools. Figure 2 summarizes these results. In addition to the used tools, we wanted to know which languages are deployed since this information could be relevant for related topics such as providing standardized data exchange. According to the survey results, the majority uses UML and/or SysML for modeling. More detailed results are presented in Figure 3. Standard Domain Specific Language (DSL) means the use of any other standardized modeling language, except for the mentioned ones, such as EAST-ADL or AUTOSAR which were commonly mentioned. It is important to note that we accidentally omitted Stateflow charts in the question. This explains the paradox that, even though Matlab tops the list in the modeling tools question, UML is ranked highest in the modeling language question. Moreover, we are interested in applied notations and diagram types as they give information about which types of models are used in practice and, hence, which aspects are important to model. Finite State Machines are ranked first by participants (73 persons/85 departments), followed by sequence-based models (62 persons/69 departments) and block diagrams (61 persons/79 departments). More details regarding the usage of the notations can be found in Figure 4. In addition to notations, we asked participants for which functional aspects they use models. Results show that the majority of all subjects use models for structure aspects (69 persons/80 departments), followed by discrete state/event based specifications (48 personal/60 department), and static interfaces (48 persons/61 departments). From there on, the usage is declining (approximately) linearly with 20 participants using models for safety aspects and 23 for hybrid behavior. Detailed results for this question are depicted in Figure 5.

While similar questions were asked in related work, e.g., [16, 9], they are difficult to compare due to their focus and age. Grossman et al. covered both the used diagram types and the used modeling tools [16], but only with respect to UML. In their case, the most commonly used UML diagram types are Use Case, Class, and Sequence Diagrams. For an overview of the different diagram types, see the official standard [30]. Similarly, Dobing and Parson's study ranks Use Case, Class, and Sequence diagrams as the most commonly used diagram types [9]. Both structural models (including UML Class diagrams) and sequence-based models are among the most commonly used diagrams in our data as well. Use Case diagrams have not been mentioned in our survey. This could be related to the fact that we did not include this diagram type as a choice in the answers (however, participants had the possibility to mention it in the free-text answers). However, not using Use Case diagrams could also indicate that there has been a paradigm shift away from the Unified Process. In the studies conducted by Grossman et al., 78.6% of the participants used the Unified Process, and regarding tooling, Rose and Enterprise Architect are by far the most commonly used UML tools [16]. This is partly reflected in our data, where Enterprise Architect and IBM Rational Software Modeler, which can be seen as the successor of Rose, are highly ranked. Interestingly, both Eclipse-based tools and in-house tools are equally common. This could be attributed to the fact that the scope of our study is not limited to UML as a modeling language. However, UML and common UML extensions, such as SysML, are still used by the majority of the participants (cf. Figure 3). Hence, it seems like there has been a shift away from commercial UML tools to the use of open-source alternatives or in-house tools.

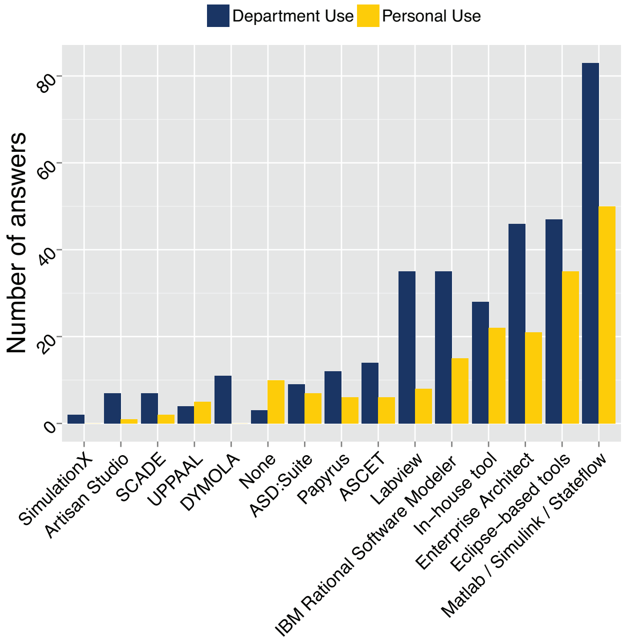

Fig. 2 Modeling tools

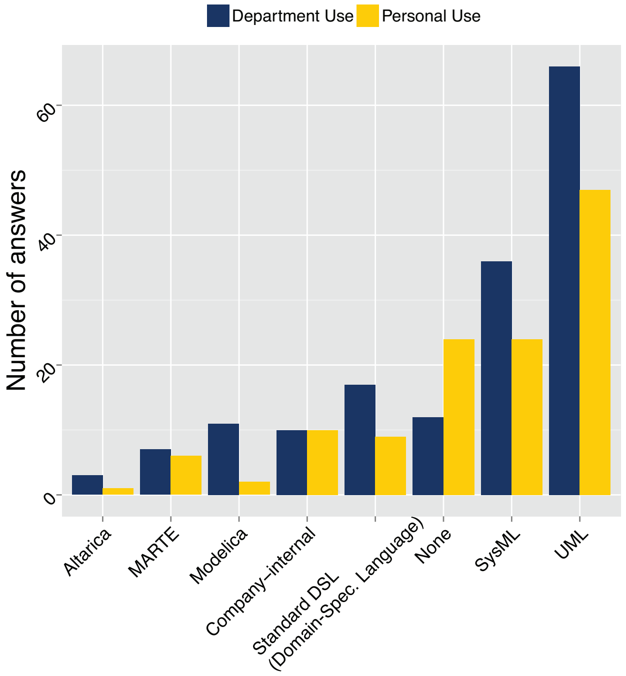

Fig. 3 Modeling languages

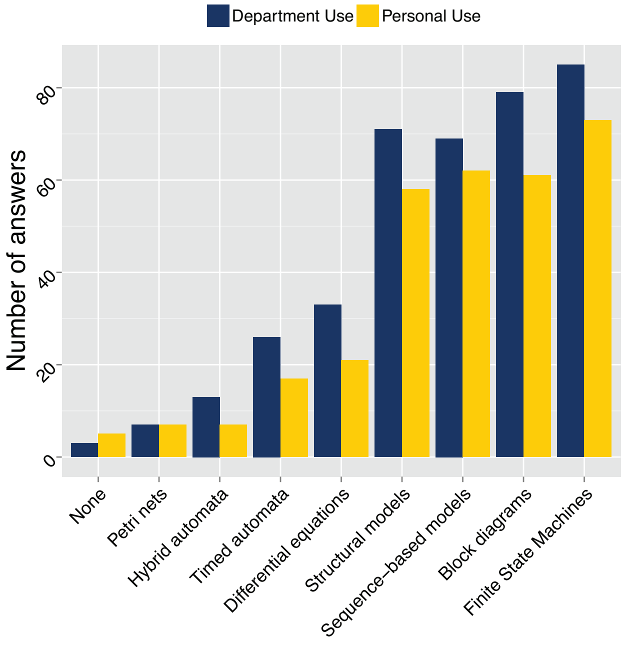

Fig. 4 Type(s) of notations

Further, we wanted to know for which purposes models are used in the divisions or departments of the participants. The results for this question are illustrated in Figure 6. According to the responses, models are mainly used for simulation, code generation, and for information/documentation; hence, the automation of activities in the development process seems to be highly important. This is in contrast to the findings of Agner et al.'s study, which reports little use of code generation [4]. In contrast, timing analysis, safety compliance checks, reliability analysis and formal verification do not seem to be used widely. The survey data does not contain any information with respect to the kind of models and modeling languages which are used for each of the stated purposes. In order to answer this question, a more detailed follow-up study would be required.

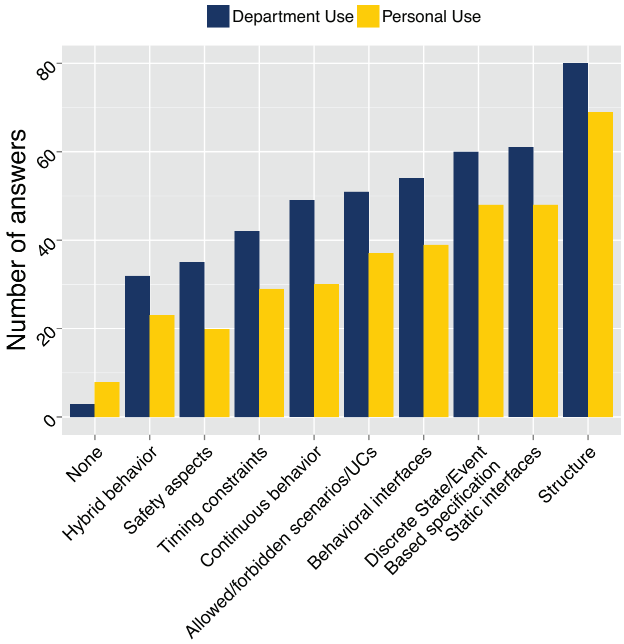

Fig. 5 Functional Aspects

The low number of participants using formal verification is both notable and interesting given the strong need for formal methods as stated in Section 4.3. We see two potential reasons for this low number. First, even though the need for formal methods is already evident in our survey data, the necessary steps to actually implement formal methods in practice might still be progressing. In their 2009 survey of formal methods in practice, Woodcock et al. argue that "the time is right to attempt to make significant advances in the practical application of formal methods and verification in industry" [37]. However, they do not confirm that the the advances have been taking place at the time. Our data indicates that this view still is held and also by practitioners in the embedded systems domain. Second, it might be that formal methods actually have been adopted within the company, but they are not in use in the particular division/department of the survey participants. Our data does not offer any strong insights into whether any of these two reasons are true.

As a typical development process includes several tools from different tool vendors, exchanging data between these tools is necessary. Depending on the tool integration mechanism, exchanging models and establishing traceability between tools can lead to substantial efforts. Therefore, we wanted to know how the data exchange is done in practice and asked participants which integration mechanisms they currently apply. From the results we see that the import/export of models using defined file formats is the most common approach (55 answers). Detailed results are depicted in Figure 7. The interoperability challenge with tools is one of the main perceived shortcomings of MBE (see Section 4.2.4), existing integration mechanisms indeed have room for improvement.

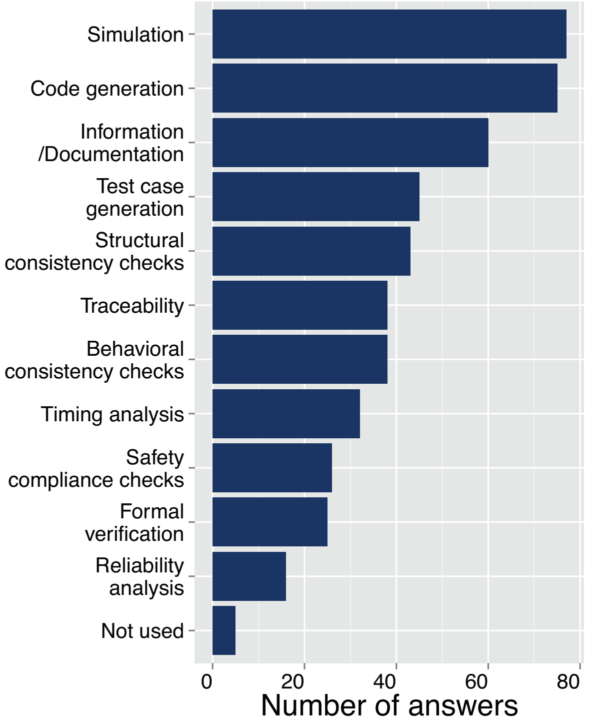

Fig. 6 Model purpose

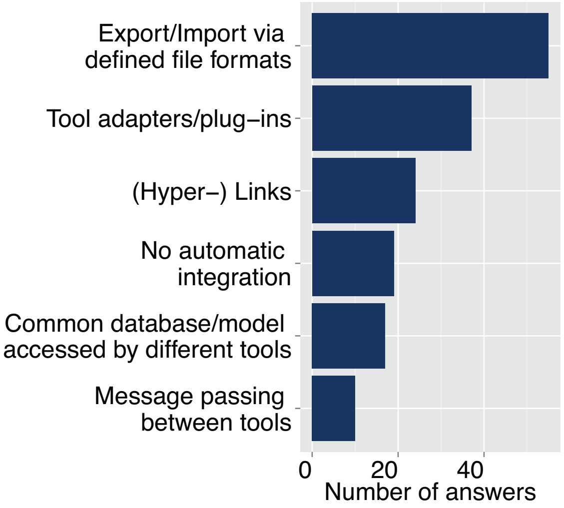

Fig. 7 Type(s) of integration mechanism(s)

4.2.2 Needs for introducing MBE An interesting issue is the motivation why companies decide to use models for developing their systems. Asking for motivations for introducing MBE can shed light and offer valuable information about companies' opinions regarding the advantages of MBE as well as challenges they are faced with and what they want to solve with MBE. Therefore, we included one question about the needs for introducing MBE in the survey. We summarize the results in Figure 8.

In the figure, the needs, which have been stated in the questionnaire, and the responses are listed. The three percentage declarations in the figure show (on the left side) the percentage of the answers with 'not relevant' and 'somewhat relevant', the percentage of the neutral 'relevant' answers (in the middle), and (on the right side) the percentage of answers with 'mostly relevant' and 'very relevant'. The figures in the following sections follow the same format as Figure 8, but the possible answers are different depending on the actual question. The figure shows that most participants (68%) think that their company adopted MBE because they had a need for shorter development time and needed to improve reusability. Further, more than 50% say that needs for quality, maintainability, and reliability improvements, as well as cost savings and traceability, are reasons for applying MBE. Least important for the respondents are the needs to improve availability and confidentiality; similarly neither customers nor standards do appear to strongly require MBE. Some participants provided free text answers to this question. Among others, 'better capturing of customer needs', 'exploration of architecture alternatives', or 'need for reducing development time by using code generation' are mentioned, but there are no common patterns in the answers.

4.2.3 Positive and negative effects of MBE In addition to the needs for introducing MBE, the effects of the actual use of MBE are interesting. There could be positive as well as negative effects; hence, we asked in a general manner "What were the effects of introducing MBE in your division/department?". Figure 9 shows the answers for this question on different aspects of systems engineering or on the product. For each aspects, between 7 and 12 people did not answer and between 30 and 51 answered 'Don't know'.

Quality, reusability, reliability, traceability, maintainability, development time, formal method adoption, integrity, safety, availability, cost, and efficiency of resulting code are rated highly or partially positive by most participants. Standard conformity and confidentiality have no effect according to more than 50% of the participants. In the free text answers, further positive effects with respect to 'customer need capturing', 'usability for simulation', and 'easy access for R&D engineers' are mentioned. One participant stated that the effects are not yet measurable; several participants stated that they do not use MBE or are only developing tools and methods and can therefore not assess the effects, at least not yet. In summary, most survey participants think that MBE has more positive than negative effects.

4.2.4 Shortcomings of MBE In order to identify potential improvements, we asked subjects about current shortcomings of MBE. Figure 10 shows the answers for this question which range from "does not apply at all" to "fully applies". Many survey participants consider existing interoperability interfaces to be one of the main shortcomings. One participant stated in the free text answer field that this is true, but only for commercial tools. Other shortcomings concerning commercial tools, which have been explained in free text answers, include the steady increase of license costs per developer and the lack of tool support and maintenance by tool vendors. This could be the reason for why companies use a lot of open source and in-house developed tools (see Section 4.2.1). Additionally, one participant states that in his/her case, all of the shortcomings have been addressed by using in-house DSLs.

4.2.5 MBE Tool Usage In order to evaluate how familiar subjects are with MBE tooling, we asked how much time they spend using MBE tools in comparison to non-MBE tools. Here, 5 subjects stated they do not use any MBE tools, 25 answered that they use fewer MBE tools than non-MBE tools, 47 use more MBE tools than non-MBE tools, and 12 use only MBE tools. Finally, 8 answered that they do not perform any engineering activities. Hence, a majority of all participants use MBE tools in their daily business.

4.2.6 Hypotheses Evaluation As discussed in Section 3, we elicited the hypotheses from related work, but independent of the design of the questionnaire. While this provides us with a base for future research in the area of MBE, it also means that several hypotheses cannot be evaluated based on our survey. This is mainly due to a lack of questions addressing these hypotheses, e.g., in the case of $H_{adv}.4-8$, $H_{chall}.4$, $H_{chall}.7$ and $H_{chall}.11$. Hypotheses $H_{chall}.9$ and $H_{adopt}.3$ cannot be evaluated as they address a different population and would not be answerable based on our sample from the embedded systems domain. Finally, we have data regarding the modeling languages, showing a strong support for UML. However, as discussed in Section 4.2.1, Matlab is the most widely used tool according to our data set. It would be logical that Stateflow charts is also one of the leading modeling languages. We later realized that we did not include Stateflow charts in the modeling language question. Therefore, we cannot evaluate $H_{adopt}.1$ without introducing a large threat to validity. The evaluation results are summarized in Table 3.

Out of the eight hypotheses targeting potential advantages (H_{adv}), four can be evaluated using our data. Based on Section 4.2.3, $H_{adv}.1$ (quality improvements)

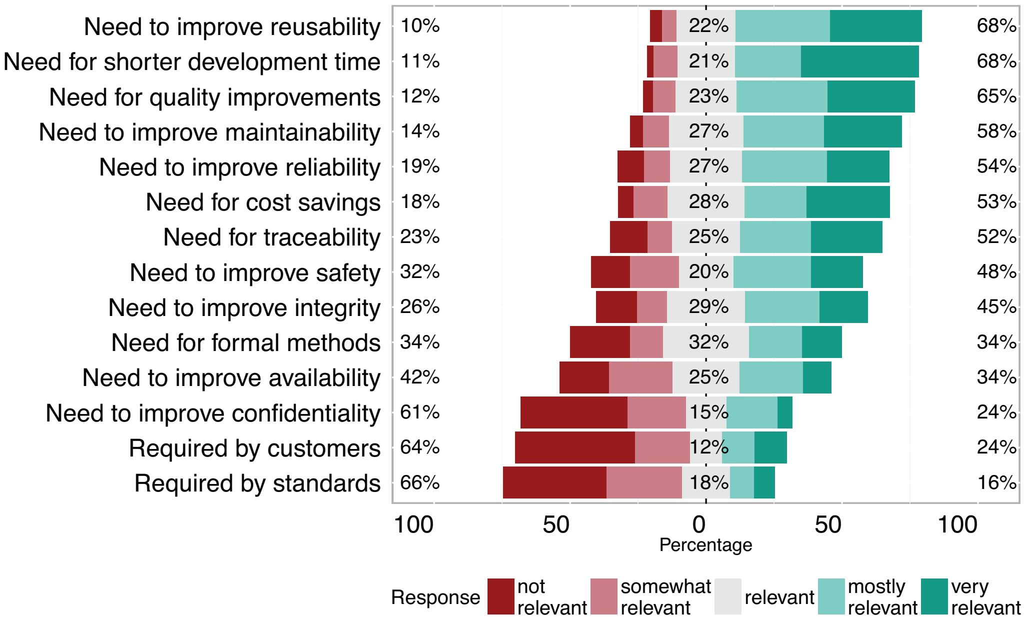

Fig. 8 Reasons for introducing MBE

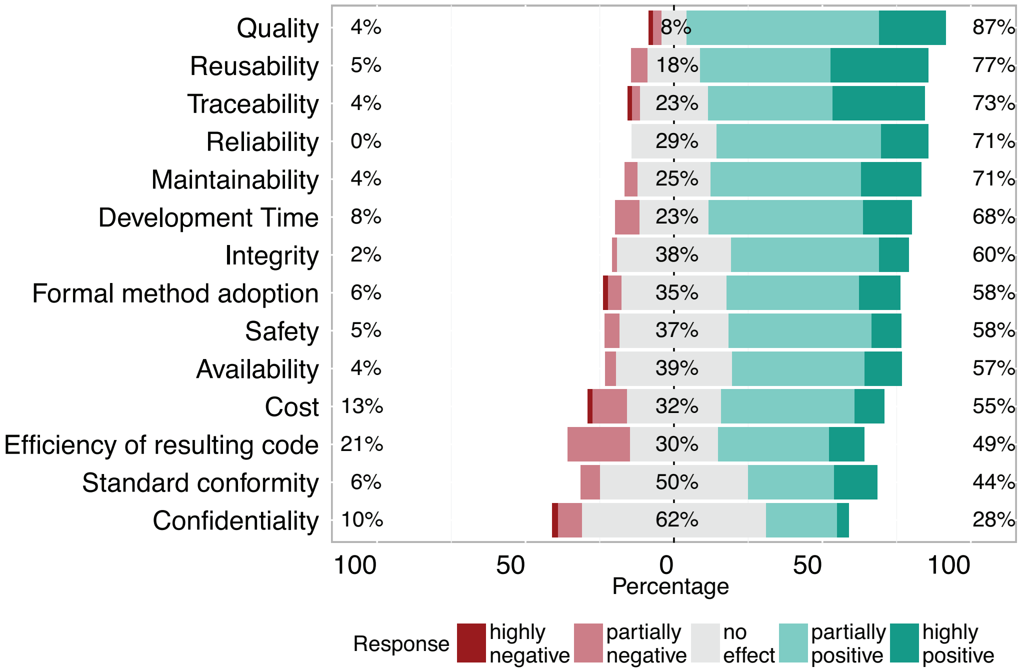

Fig. 9 Positive and negative effects of MBE

is supported by the data. $H_{adv}.2$ (productivity improvements) and $H_{adv}.6$ (increased development speed of new requirements) are supported with respect to development times. Other aspects of these hypotheses, such as productivity improvements due to increased efficiency, are not captured by our questionnaire. Finally, $H_{adv}.3$ is supported with respect to maintainability.

Our data on shortcomings of MBE presented in Section 4.2.4 can be used to evaluate a number of hypotheses regarding MBE challenges ($H_{chall}.X$) from our list. Regarding tool integration, $H_{chall}.5$ is clearly supported.

Moreover, more than one third of the people think that MBE requires a high effort to train developers (supporting $H_{chall}.3$), that there are usability issues with tools (supporting $H_{chall}.2$ with respect to usability) and

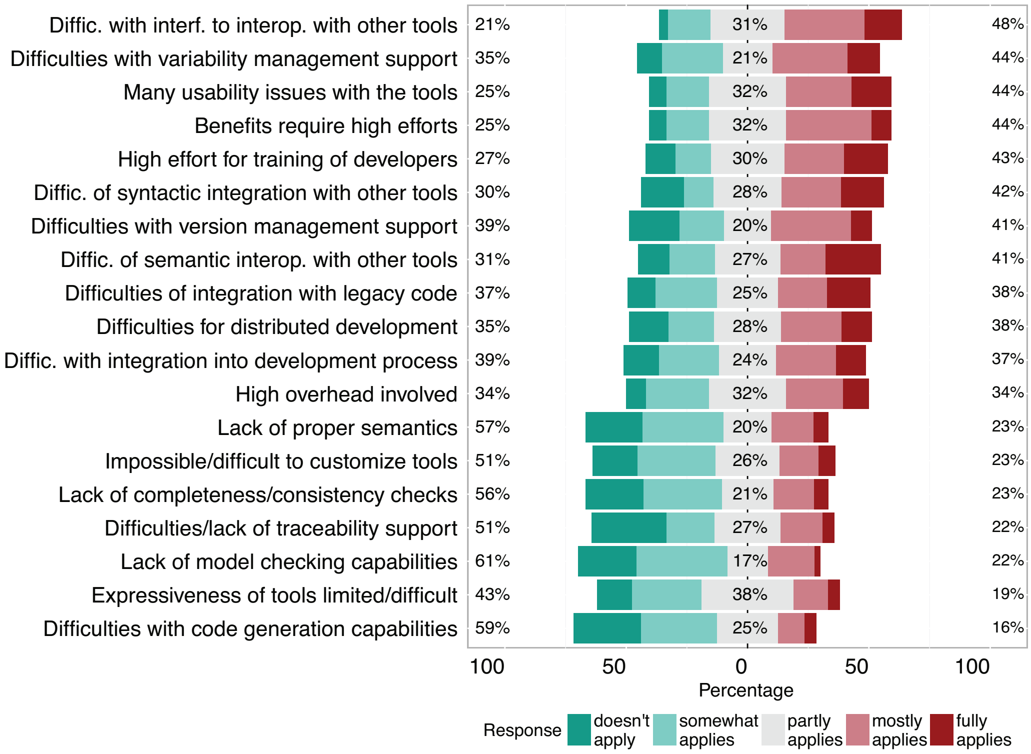

Fig. 10 Shortcomings of MBE

that benefits require high efforts (supporting $H_{chall}.10$ and supporting $H_{chall}.12$ with respect to the required effort). Even though $H_{chall}.10$ is supported by "benefits require high efforts", opinions about whether high overhead is involved with the usage of MBE vary.

No shortcomings according to the responses are difficulties to customize tools and limitations on what can be expressed within tools, which is opposing $H_{chall}.2$ with respect to customization aspects and expressiveness. Hence, although the interoperability between tools seems to be the main shortcoming, capabilities of single methods and tools are satisfactory for many surveyed subjects. According to the majority of the participants, there are no shortcomings with respect to code generation capabilities in MBE, indicating that $H_{chall}.6$ does in fact not apply in our setting. While "High effort for training developers" is considered to be one of the main shortcomings by the participants, it is unclear if this is due to the complexity of modeling in general, which would then support $H_{chall}.8$. Hypothesis $H_{chall}.1$ can only be evaluated with respect to legacy code. Here, 38% of the participants indicated that using MBE with legacy code is challenging.

Several of the other hypotheses regarding adoption of MBE can be answered using the survey data. Most people in our sample use MBE, not only participants who work for large companies. In fact, out of the 5 participants who state that they do not use MBE tools, 3 work at large companies. This means that $H_{adopt}.2$ is false in our data.

As discussed in Section 4.2.1, there is a substantial use of code generation, opposing $H_{adopt}.4$.

Finally, our data shows that a large proportion of participants who use code generation do not use Matlab, thus refuting $H_{adopt}.5$. Only 38 participants out of 75, who indicated that they use code generation, do use Matlab. Among the remaining 37, Eclipse-based and in-house tools are the most popular.

Compared to related work, the differences in our data could be related to the nature of the domain we are studying. Circumstances such as safety standards could require, or at least encourage, modeling within the embedded systems domain. This would explain why models are generally widely used, not only in larger companies ($H_{adopt}.2$). Similarly, this could explain why code generation is widely used ($H_{adopt}.4$) and not only by Matlab users ($H_{adopt}.5$).

Summing up, it can be said that many survey participants think that the positive effects outweigh the negative effects of MBE. However, the interoperability between tools and the usability of them, the effort to train developers, as well as that the benefits require high efforts are considered as the main shortcomings of MBE.

Table 3 Evaluation of Hypotheses from Related Work

Hypothesis	Description	Supported
$H_{adv}.1$	MBE leads to a reduction of defects/improvements in quality.	Yes
$H_{adv}.2$	MBE leads to improvements in productivity.	Partially
$H_{adv}.3$	MBE increases understandability.	Partially
$H_{adv}.4$	Advantages of MBE are simulation and testing, and performance-related decision support.	No data
$H_{adv}.5$	MBE leads to an earlier detection of errors.	No data
$H_{adv}.6$	MBE can speed up the implementation of new requirements.	No data
$H_{adv}.7$	Code generation is an important aspect of MBE productivity gains.	No data
$H_{adv}.8$	MBE leads to a higher degree of automation.	No data
$H_{chall}.1$	Using MBE with legacy systems is challenging.	Partially
$H_{chall}.2$	Current MBE tools are insufficient.	Yes/No
$H_{chall}.3$	Significant additional training is needed for using MBE.	Yes
$H_{chall}.4$	Managing models of complex systems is challenging.	No data
$H_{chall}.5$	Tool integration is challenging.	Yes
$H_{chall}.6$	Code generated from models has poor performance.	No
$H_{chall}.7$	MBE lacks scalability.	No data
$H_{chall}.8$	The complexity of modeling is challenging.	Partially
$H_{chall}.9$	Companies which consider software development their main business seem to find the adoption of MBE more challenging than other companies.	No data
$H_{chall}.10$	Modeling requires too much effort.	Yes
$H_{chall}.11$	Handling the consistency of models over time is challenging.	No data
$H_{chall}.12$	Modeling is not useful enough.	Partially
$H_{adopt}.1$	UML is the preferred modeling language employed in MBE.	No data
$H_{adopt}.2$	Models are used mainly in larger companies.	No
$H_{adopt}.3$	UML is mostly used by experienced developers working at medium-sized companies.	No data
$H_{adopt}.4$	There is little use of code generation or model-centric approaches.	No
$H_{adopt}.5$	Code generation is mainly used by subjects using Matlab/Simulink/Stateflow.	No

4.3 RQ2: Differences by Subgroups

4.3.1 Demographic Subgroups In the following paragraphs, we discuss potential differences between subgroups of survey participants with respect to survey questions mainly concerning the assessment of MBE. Nevertheless, the actual SoP in the subgroups is analyzed and described as well in cases where we observed discrepancies. We address domains, positions in the value chain, company size, and product size targeted by the participants' employers, for this purpose. Though there are several possibilities for analysis of discrepancies, we have selected these four subclasses because we think that differences between these subgroups are most likely.

Domain The domains which we analyze within this subgroup are Automotive, Avionics, Healthcare, and Defense. Results of other participating domains are not representative due to too few answers rendering analysis not being viable.

For all analyzed domains Matlab is the most used tool, whereas the successive frequent usage of modeling environments differs slightly. Mostly Avionics, but also Healthcare and Defense, use Rational Rhapsody proportionally more than the Automotive domain, which instead shows an increased usage of Enterprise Architect. Generally, participants from Automotive, Avionics, and Defense domain state that they use more MBE tools than non-MBE tools. According to the number of answers, the Healthcare domain uses less MBE tools than non-MBE tools. Although the top answers for the purpose of models are similar for all domains, there are some differences in the ranking. Automotive uses models mainly for simulation, whereas Avionics uses them nearly equally often for information and documentation, simulation, and code generation. Healthcare applies models comparably often for simulation and code generation, whereas the main purpose of models in the Defense domain is information/documentation followed by code generation. Phases in which models are applied differ slightly. In the Automotive domain, models are applied models mainly for implementation whereas the other domains apply models mostly for specifying the system architecture. Additionally, the Healthcare domain applies models for subsystem and component design, and the Defense domain for subsystem and component test equally often as for system architecture.

The answers for the assessment of MBE are comparable to the overall outcome whereas the order can differ between domains. Table 4 summarizes the three answers which received the highest ratings by participants for each analyzed aspect. The depicted percentages refer to the percentage of participants that answered 'very relevant' or 'mostly relevant' for the introduction needs, 'highly positive' or 'partially positive' for positive effects, 'highly negative' or 'partially negative' for introducing MBE, and 'fully applies' or 'mostly applies' for the shortcomings of MBE. Generally, the participants see very few negative effects of MBE and hence, the results are based only on a few answers. As answers sometimes have the same percentage, such as development time, reusability, and quality as positive effects in the automotive domain, several properties can be found in one cell. In cases where we do not have enough answers, the table cell contains 'na' (not available). This is the case when many participants answered 'don't know' or did not answer the question at all. Though the ranking of assessed attributes differs slightly between domains, no strong discrepancies are found in the answers. Participants of all domains think that MBE has a positive effect on quality and, except for Defense domain, also on development time and traceability. The Defense community notably rates development time as a negative effect of MBE (second most common answer). However, negative effects for which all subgroups agree are cost and code efficiency. More details can be found in Table 4.

Thus, all in all there are no large differences between domains.

Position in Value Chain Participants from different value chain positions overall agree in answers; only small differences can be found. Enterprise Architect is more used at second-tier suppliers, in fact equally often as Matlab. First-tier and second-tier suppliers also indicate that finite state machines and timed automata are the most used modeling notations whereas OEMs mainly use finite state machines and structural models. All subgroups use models mainly for describing structural aspects followed by static interfaces at OEMs, discrete state/event-based specifications at first-tier suppliers, and behavioral interfaces at second-tier suppliers. According to the results, OEMs use MBE tools less than non-MBE tools in contrast to first-tier suppliers, which use more MBE tools, and second-tier suppliers, where both are used equally often. The main purpose of models at OEMs is for capturing information/documentation in contrast to the other subgroups where the main function of models is simulation, or simulation and code generation, respectively, at first-tier suppliers. Further, OEMs and first-tier suppliers mainly exchange data via defined file formats in contrast to second-tier suppliers, which use tool adapters more often. MBE is mainly applied for subsystem and component design, and system architecture at OEMs. For first-tier suppliers, implementation, and subsystem and component design are most common; among second-tier suppliers, system architecture, subsystem and component design, implementation, and subsystem and component test receive most answers.

In table 5, the highest rated answers for the assessment of MBE are summarized following the same structure as for the domain subgroups. The table shows that positive effects predominate negative effects for all subgroups. All report positive effects on quality, and negative effects on cost/code efficiency. However, there are also reported discrepancies of positive and negative effects for one subgroup. Second-tier suppliers rank reusability in third position for both positive (79%) and negative (7%) effects. This has to do with the fact that participants do not see many negative effects of MBE in general and, hence, reusability is ranked in third position for the negative effects, although only 7% of participants think that MBE has a negative effect on it. In summary, the position in the value chain does not seem to strongly affect the view on RQ1.

Company Size The data analysis for the three subgroups based on company size, namely SME, Large Company, and University/Research Institute, do not show significant differences in answers, and in that sense it is similar to the previously described subgroups. However, it has to be taken into account that most of the participants work for large companies (78%) (see Section 4.1). SMEs and research institutes use Eclipse-based tools nearly equally as much as Matlab. In contrast, participants working at large companies use Matlab twice as much as Eclipse. Further, SMEs use less differential equations and more non-MBE tools than MBE tools compared to the other subgroups. The main applied notation type at large companies are state charts whereas SMEs apply structural models most often. SMEs mostly use models for behavioral consistency checks, large companies for code generation, and universities/research institutes for simulation.

Table 6 summarizes the answers for this subgroup regarding the assessment of MBE. It shows that 29% of university participants think that MBE has a partially negative effect on safety, compared to only 2% of participants at large companies, and 0% of participants at SMEs. Further, 67% of subjects at universities/research institutes introduced MBE because of a need for formal methods but only 30% of SMEs and large companies adopted it for this reason. Nevertheless, it has to be stated that we only have few answers from the university subgroup, e.g., only 4 answers for "lack of semantic interoperability" and 3 for "lack of syntactic integration". However, though the ranking of answers differs between subclasses, no major contradictions exist for this subgroup either.

Target Product Subgroups offering different target products do also not show considerable differences in their answers. We analyzed differences for commercial products produced in large scale and in medium scale, as well as products developed as research demonstrators. For the other subclasses, small scale producers and non-commercial products, we do not have enough data. One notable aspect is that participants using MBE for large-

Table 4 Answers grouped by different domains

	Introduction needs		Positive effects		Negative effects		Shortcomings	
Automotive	Development Time	73%	Development Time /Quality/Reusability	84%	Code Efficiency	25%	Interoperability Interfaces	45%
	Reusability	68%	Traceability	83%	Cost	14%	Semantic Interop. /Variability Mgmt	44%
	Maintainability	63%	Reliability	74%	Confidentiality /Availability	8%	Tool Usability	43%
Avionics	Reusability	83%	Quality	90%	Code Efficiency	19%	Interoperability Interfaces	57%
	Quality	71%	Formal Method Adoption	74%	Development Time	15%	Tool Usability	56%
	Development Time/ Reliability	62%	Reusability	72%	Cost	10%	Syntactic Integration	55%
Healthcare	Quality	72%	Quality	88%	Cost	29%	Distributed Development	62%
	Development Time	61%	Maintainability/ Traceability	69%	Code Efficiency	25%	Interop. Interfaces/ Training Efforts	56%
	Maintainability	60%	Development Time /Reusability/Reliability	56%	Development Time	17%	High Efforts	53%
Defense	Quality	75%	Traceability	100%	Code Efficiency	33%	Version Management	64%
	Reusability	73%	Development Time	90%	Cost	10%	Tool Usability	58%
	Traceability	70%	Quality	83%	na		Interop. Interfaces /Consistency Checks	50%

Table 5 Answers grouped by different value chain parties

	Introduction needs		Positive effects		Negative effects		Shortcomings	
OEM	Reusability	70%	Quality	81%	Cost	22%	Tool Usability	52%
	Development Time	63%	Reusability	71%	Code Efficiency	17%	Interoperability Interfaces	51%
	Reliability	60%	Maintainability	69%	Confidentiality	16%	Distributed Development	46%
1st Level Supplier	Quality	72%	Quality	88%	Code Efficiency	21%	Tool Usability	58%
	Development Time	71%	Traceability	81%	Cost	11%	High efforts /Interop. Interfaces	49%
	Reusability	66%	Reliability	80%	Formal Method Adoption	10%	High Training Effort	47%
2nd Level Supplier	Development Time	68%	Quality	93%	Cost	18%	Semantic Interoperability	56%
	Quality	67%	Maintainability/ Traceability	82%	Code Efficiency /Development Time	8%	Interoperability Interfaces	55%
	Maintainability	60%	Reusability	79%	Reusability	7%	High Efforts	53%

scale productions and demonstrators use Matlab more in contrast to medium-scale producers which use Matlab, Enterprise Architect, and Eclipse tools comparably often. Furthermore, large-scale developers and demonstrator developers bear resemblance to each other also regarding the used modeling notations and the purpose of models. Both use finite state machines and block diagrams most often in contrast to medium-scale producers which use more other structural models and sequence-based notations. Their main purpose of models is simulation whereas medium-scale product developers make use of code generation from models more often.

Table 7 lists the number of answers for the analyzed target product subgroups. The table shows that all analyzed groups needed shorter development times and introduced MBE for this reason. Furthermore, participants from these subgroups see different MBE effects and shortcomings, at least in the top answers (see Table 7). There is actually a discrepancy between large-scale producers, which see positive effects on development time, and medium-scale producers, where development time is

Table 6 Answers grouped by different company sizes

	Introduction needs		Positive effects		Negative effects		Shortcomings	
University	Reusability	80%	Reusability	100%	Confidentiality	33%	Semantic Interop./ Syntactic Integration	100%
	Costs	71%	Availability/Integrity /Maintainability	83%	Safety	29%	Variability Management /High Overhead	67%
	Reliability/Safety	70%	Formal Method Adoption	75%	Quality	20%	Integration Legacy Code/High Efforts	50%
SME	Reliability/Quality	73%	Quality /Traceability	90%	Confidentiality	29%	Legacy Integration	70%
	Development Time	67%	Reusability/Reliability /Maintainability	80%	Code Efficiency	22%	High Training Effort	67%
	Maintainability	60%	Development Time	78%	Standard Conformity	12%	High Overhead	56%
Large Company	Development Time	69%	Quality	89%	Code Efficiency	23%	Interoperability Interfaces	50%
	Reusability	68%	Reusability	73%	Cost	13%	Tool Usability	45%
	Quality	65%	Traceability	71%	Development Time	11%	Variability Management	44%

ranked second for negative effects, with 20% assessing it negatively.

All in all, no essential differences between the analyzed subgroups could be detected. In this paper, it should be noted, we only analyzed subgroups where we expected some differences with a high probability. For further analysis, we provide the raw data as described in Section 3.

4.3.2 Hypotheses Evaluation In the following, we discuss the results on **RQ2** with respect to our hypotheses about differences in answers of subgroups of survey participants. As shown in the previous section, the answers of the survey participants are ordinal scaled, e.g., a likert scale for the question about positive and negative effects of MBE. Thus, we have to use a statistical test which supports ordinal scaled data to assess if the differences are significant. We use Fisher's exact test [12] (two-tailed) with a level of significance $\alpha \leq 0.05$. This test is a non-parametric statistical test for contingency tables. In our case, the contingency table consists of the answers of the participants in the columns and the different subgroups in the rows. The evaluation results are summarized in Table 8.

The hypotheses **H2.1**, **H2.2**, **H2.3**, **H2.4**, and **H2.5** address the full list of positive and negative effects as presented in Section 4.2.3. We check and report significance for each effect (e.g., cost and quality) individually.

For hypothesis **H2.4**, we do not have enough data for each subgroup in order to compare the groups. Hypotheses **H2.1** and **H2.7** did not show any significant differences (i.e., $p \geq 0.05$) between the subgroups. Hence, here we cannot reject the null hypotheses.

With respect to Hypothesis **H2.2**, traceability ($p \approx 0.000027$), safety ($p \approx 0.014$), and reusability ($p \approx 0.012$) yielded significant differences. That is, supporters of MBE perceive the effects of MBE on these three aspects significantly more positive than subjects who are opposed to or neutral towards MBE (See Fig. 9 for the complete sample). On traceability, 84% of MBE supporters report partially or highly positive effects, in contrast to only 27% for the opponents and neutral participants. Note that in our sample there is only one opponent of MBE.

For Hypothesis **H2.3**, that is participants who still use MBE see more positive effects of MBE than participants who stopped using MBE, we get significant differences for cost ($p \approx 0.027$) and traceability ($p \approx 0.012$). That means that participants who are still using MBE report in total more positive effects on cost and traceability than participants who stopped using MBE. For instance, 81% of the participants still using MBE report partially or highly positive effects on traceability whereas only 52% of the participants who stopped using MBE report it. A possible explanation for the few significant differences might be that participants who stopped using MBE did so because they moved to a different position, e.g., in management, and not because they did not see the benefits of MBE.

Investigating whether subjects who see many usability issues with MBE tools also report more negative effects than other subjects (**H2.5**), we only observe a significant difference with respect to quality ($p \approx 0.0079$). Participants, who reported that many usability issues with tools mostly or fully applies, rated the effects on quality slightly less positive (10% highly or partially negative, 14% no effect, and 76% partially or highly positive) than participants who reported that usability issues apply at most partially (0%, 5% and 95%).

Apart from being insufficient, the complexity of MBE tools is often mentioned as a challenge. This would lead to the impression that once you are experienced in MBE, it should be easier to use these tools. Hence, we derived

Table 7 Answers grouped by different target products

	Introduction needs		Positive effects		Negative effects		Shortcomings	
Research Demon-strator	Reliability	76%	Reusability	92%	Cost	20%	Syntactic Integration	85%
	Development Time	75%	Reliability/Formal Method Adoption	91%	Confidentiality	17%	Interoperability Interfaces	71%
	Quality	73%	Integrity/Availability	88%	Standard Conformity /Code Efficiency	11%	Variability Management	62%
Medium Scale Product	Development Time	81%	Quality	100%	Cost	23%	High Training Effort	71%
	Quality	80%	Maintainability	87%	Development Time	20%	Tool Usability	67%
	Reliability	73%	Reliability	77%	Confidentiality	15%	Tool Customization	64%
Large Scale Product	Development Time	74%	Quality	88%	Code Efficiency	33%	Interop. Interfaces /Variability Management	44%
	Reusability	70%	Development Time	82%	Maintainability /Availability	10%	High Efforts	42%
	Maintainability	70%	Reusability	81%	Formal Method Adoption	8%	Semantic Interoperability	41%

the hypothesis that experienced MBE users report less problems than MBE novices (**H2.6**). For this hypothesis, we only observe a significant difference in our data with respect to a lack of proper semantics ($p \approx 0.0089$).

Supporters of MBE also use more MBE tools in comparison to subjects who are opposed to or neutral towards MBE (**H2.8**) ($p \approx 0.000083$, less-than Fisher test). Here, 53 supporters of MBE reported to use MBE tools more than non-MBE tools or only MBE tools, and 16 reported to use less MBE tools than non-MBE tools or no MBE tools at all. This contrasts with a score of 5 and 14 answers on the opponent/neutral side.

In total we performed 72 significance checks resulting in eight significant differences. Though the number of found significances is low for this amount of significance checks, we believe that our results could be used as indicators for future studies.

5 Conclusions and Future Work

While Model-Based Engineering has a long history in the embedded systems domain, it is unclear to which extent Model-Based Engineering is used in industry today. Especially empirical data on the adoption, advantages, and disadvantages of Model-Based Engineering is scarce from this domain. In this paper, we presented our results from a survey on Model-Based Engineering in the embedded systems domain. We collected data from 113 participants, of which 102 are working in industry. The presented results strongly confirm that indeed Model-Based Engineering is widely used in the embedded systems domain. Models are clearly not only used for informative and documentation purposes; they are key artifacts of the development processes, and they are used for, e.g., simulation and code generation. Other widespread uses of significant importance are behavioral and structural consistency checking, as well as test case generation, traceability and timing analysis. Common modeling tools are Matlab/Simulink and Eclipse-based tools, with UML and SysML being the leading modeling languages. While survey respondents reported mostly positive effects of Model-Based Engineering, the data also suggests some common and major challenges that need further attention. These challenges include effective adoption among developers to reduce effort-intensive activities currently needed to realize benefits of Model-Based Engineering, as well as several challenges with respect to tooling, such as interoperability and usability. Our data shows only small differences between different subgroups, e.g. different domains, positions in the value chain, or company sizes. This indicates that the embedded systems domain is in fact having similar experiences and practices, and is bearing similar challenges with respect to Model-Based Engineering.

In the future, we plan on following up the results of this study by replicating the survey with a different target group in the embedded domain to validate the identified results. Furthermore, a validation of some effects of the introduction of Model-Based Engineering can be performed by collecting quantitative data in a company which introduces a Model-Based Engineering approach.

Acknowledgements The research leading to these results has received partial funding from the European Union's Seventh Framework Program (FP7/2007-2013) for CRYSTAL-Critical System Engineering Acceleration Joint Undertaking under grant agreement No 332830 and from Vinnova under DI-ARIENR 2012-04304. Further, the authors gratefully acknowledge financial support from FFG Austria for the project in which the above presented research results were achieved. We would also like to thank Aleksander Lodwich, Jos Langen and Leon Bouwmeester for participating in the survey design and contributing to the final study.

Table 8 Evaluation of Hypotheses for RQ2

Hypothesis	Description	Significant Differences
H2.1	Users of in-house tools report more positive and less negative effects of MBE than users who do not use in-house tools.	None
H2.2	Supporters of MBE report more positive effects than subjects opposed to or neutral towards MBE.	Traceability ($p < 0.01$) Safety ($p \approx 0.014$) Reusability ($p \approx 0.012$)
H2.3	Subjects who are still using MBE report more positive and less negative effects than subjects who stopped using MBE.	Cost ($p \approx 0.027$) Traceability ($p \approx 0.012$)
H2.4	Subjects who only use models for means of information/documentation report less positive than negative effects.	Not enough data
H2.5	Subjects who do not see many usability issues with MBE tools report fewer negative effects.	Quality ($p < 0.01$)
H2.6	Highly experienced users of MBE report less problems with MBE tools than users with less experience.	Lack of proper semantics ($p < 0.01$)
H2.7	Large companies have more tool integration problems than SMEs.	None
H2.8	MBE promoters use more MBE tools in comparison to subjects neutral or opposed to MBE.	Yes ($p < 0.01$)

References

1. IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990 pp. 1–84 (1990)
2. 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)
3. 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 (2012)
4. Agner, L.T.W., Soares, I.W., Stadzisz, P.C., Simão, J.M.: A brazilian survey on UML and model-driven practices for embedded software development. Journal of Systems and Software **86**(4), 997 – 1005 (2013). SI : Software Engineering in Brazil: Retrospective and Prospective Views
5. Anda, B., Hansen, K., Gullesen, I., Thorsen, H.K.: Experiences from introducing uml-based development in a large safety-critical project. Empirical Software Engineering **11**(4), 555 – 581 (2006)
6. Baker, P., Loh, S., Weil, F.: Model-Driven Engineering in a Large Industrial Context – Motorola Case Study. In: Proc. of ACM/IEEE 8th International Conference On Model Driven Engineering Languages And Systems (2005)
7. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice. Morgan & Claypool Publishers (2012)
8. CRYSTAL: Crystal - critical system engineering acceleration. `http://www.crystal-artemis.eu` (2014)
9. Dobing, B., Parsons, J.: Dimensions of uml diagram use: A survey of practitioners. Journal of Database Management **19**(1), 1 – 18 (2008)
10. Ebert, C., Jones, C.: Embedded software: Facts, figures, and future. Computer **42**(4), 42–52 (2009)
11. Espinoza, H., Cancila, D., Selic, B., Gérard, S.: Challenges in combining sysml and marte for model-based design of embedded systems. In: Model Driven Architecture - Foundations and Applications, *Lecture Notes in Computer Science*, vol. 5562, pp. 98–113 (2009)
12. Fisher, R.A.: On the interpretation of χ^2 from contingency tables, and the calculation of p. Journal of the Royal Statistical Society **85**(1), 87 – 94 (1922)
13. Forward, A., Lethbridge, T.C.: Problems and opportunities for model-centric versus code-centric software development: A survey of software professionals. In: Proc. of 2008 International Workshop on Models in Software Engineering (MiSE) (2008)
14. France, R., Rumpe, B.: Model-driven development of complex software: A research roadmap. In: 2007 Future of Software Engineering, FOSE '07, pp. 37–54 (2007)
15. Glinz, M.: A glossary of requirements engineering terminology. Standard Glossary of the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version **1** (2011)
16. Grossman, M., Aronson, J.E., McCarthy, R.V.: Does UML make the grade? insights from the software development community. Information and Software Technology **47**(6), 383 – 397 (2005)
17. Howe, D.: Free Online Dictionary of Computing. http://foldoc.org/
18. Hutchinson, J., Rouncefield, M., Whittle, J.: Model-driven engineering practices in industry. In: Proc. of 33rd International Conference on Software Engineering (ICSE) (2011)
19. Hutchinson, J., Whittle, J., Rouncefield, M.: Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. Science of Computer Programming **89, Part B**(0), 144 – 161 (2014). SI: Success Stories in Model Driven Engineering
20. Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S.: Empirical assessment of mde in industry. In: Proc. of 33rd International Conference on Software Engineering (ICSE) (2011)
21. Karsai, G., Neema, S., Sharp, D.: Model-driven architecture for embedded software: A synopsis and an example. Science of Computer Programming **73**(1), 26 – 38 (2008). Special Issue on Foundations and Applications of Model Driven Architecture (MDA)

22. Kirstan, S., Zimmermann, J.: Evaluating costs and benefits of model-based development of embedded software systems in the car industry–results of a qualitative case study. In: Proc. of Workshop "From code centric to model centric: Evaluating the effectiveness of MDD (C2M:EEMDD)" (2010)
23. Leiner, D.J.: Too fast, too straight, too weird: Post hoc identification of meaningless data in internet surveys. `http://www.researchgate.net/publication/258997762_Too_Fast_too_Straight_too_Weird_Post_Hoc_Identification_of_Meaningless_Data_in_Internet_Surveys` (2013)
24. Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J.: Assessing the state-of-practice of model-based engineering in the embedded systems domain. In: Proc. of ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (2014)
25. Marko, N., Liebel, G., Sauter, D., Lodwich Aleksander Tichy, M., Leitner, A., Hansson, J.: Model-based engineering for embedded systems in practice. Tech. rep., University of Gothenburg (2014)
26. Mohagheghi, P., Dehlen, V.: Where Is the Proof? – A Review of Experiences from Applying MDE in Industry. In: Proc. of 4th European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA) (2008)
27. Mohagheghi, P., Gilani, W., Stefanescu, A., Fernandez, M., Nordmoen, B., Fritzsche, M.: Where does model-driven engineering help? experiences from three industrial cases. Software & Systems Modeling **12**(3), 619 – 639 (2013)
28. Niesel, K.: Volvo Car Group, Software center Day Skåne. http://www.software-center.se/digitalAssets/1506/1506474_sw-center—sk–ne-volvo.pdf (2015)
29. Object Management Group: UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems. http://www.omg.org/spec/MARTE/1.1/PDF (2011)
30. Object Management Group: OMG Unified Modeling Language 2.5. http://www.omg.org/spec/UML/2.5/PDF (2015)
31. Pfleeger, S.L., Kitchenham, B.A.: Principles of survey research: Part 1: Turning lemons into lemonade. SIGSOFT Software Engingeering Notes **26**(6), 16 – 18 (2001)
32. Punter, T., Ciolkowski, M., Freimut, B.G., John, I.: Conducting on-line surveys in software engineering. In: Proc. of the 2003 International Symposium on Empirical Software Engineering (ISESE) (2003)
33. Stachowiak, H. (ed.): Allgemeine Modelltheorie. Springer (1973)
34. Torchiano, M., Tomassetti, F., Ricca, F., Tiso, A., Reggio, G.: Preliminary findings from a survey on the md* state of the practice. In: Proc. of 2011 International Symposium on Empirical Software Engineering and Measurement (ESEM) (2011)
35. Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R.: Industrial adoption of model-driven engineering: Are the tools really the problem? In: Proc. of ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems (2013)
36. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers (2000)
37. Woodcock, J., Larsen, P.G., Bicarregui, J., Fitzgerald, J.: Formal methods: Practice and experience. ACM Computing Surveys **41**(4), 19:1 – 19:36 (2009)

A Key Terms and Definitions

Throughout the questionnaire, we use numerous terms and keywords. Even though most of these are common language in practice, we define what we mean by them in order to support readers who are not familiar with them. We reuse existing definitions from well-known sources where applicable.

A.1 Completeness

“The degree to which the specification contains all information which is necessary for developing a system that satisfies the stakeholders’ desires and needs.” [15].

A.2 Consistency

“The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a system or component.” [1].

A.3 Embedded Systems

Embedded systems are systems that are “integral components of larger systems”, which are used to “control and/or directly monitor that system using special hardware devices” [2].

A.4 Formal Methods

“Mathematically based techniques for the specification, development and verification of software and hardware systems.” [17]. This definition includes more specific terms, such as formal verification.

A.5 Formal Verification

“A functional verification process in which analysis of a design and a property yields a logical inference about whether the property holds for all behaviors of the design. If a property is declared true by a formal verification tool, no simulation can show it to be false. If the property does not hold for all behaviors, then the formal verification process should provide a specific counterexample to the property, if possible.” [3]

A.6 Integrity

“The degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data.” [1].

A.7 Model

We use the following definition, based on Stachowiak’s features of a model [33]: *A model is a representation of entities and relationships in the real world with a certain correspondence for a certain purpose.*

A.8 Model-Driven Development

“Model-Driven Development (MDD) is a development paradigm that uses models as the primary artifact of the development process.” [7].

A.9 Model-Driven Engineering

“Model-Driven Engineering (MDE) would be a superset of MDD because, as the E in MDE suggests, MDE goes beyond of the pure development activities and encompasses other model-based tasks of a complete software engineering proces.” [7]. This means that the term MDE encompasses all artifacts arising from the engineering process as artifacts that are possibly modeled, e.g. requirements, documentation, or designs.

A.10 Model-Based Engineering

The “Model-Based Engineering (MBE) process is a process in which software models play an important role although they are not necessarily the key artifacts of the development (i.e., they do NOT ’drive’ the process as in MDE).” [7].

A.11 Quality

“(1) The degree to which a system, component, or process meets specified requirements. (2) The degree to which a system, component, or process meets customer or user needs or expectations.” [1].

A.12 Simulation

“(1) A model that behaves or operates like a given system when provided a set of controlled inputs; (2) the process of developing or using a model as in (1)” [1].