Requirements for a Model-based Requirements Engineering Tool for Embedded Systems: Systematic Literature Review and Survey

Sabine Teufl, Maged Khalil, Dongyue Mou
fortiss GmbH, An-Institut an der Technischen Universität München
Guerickestr. 25, 80805 München, Germany
Email: {teufl, khalil, mou}@fortiss.org

Abstract—Model-based Requirements Engineering (MBRE) promises Requirements Engineering (RE) experts to cope with the increasing complexity of the embedded systems domain.

A major obstacle for using models more intensively in RE is the lack of appropriate tool support. A number of industrial and academic RE tools exist that support model-based development. Nevertheless, a precise set of core requirements that must be met such a tool is necessary but not yet determined. The identification of this core set of requirements for MBRE tools for embedded systems is the motivation behind the study presented in this paper.

Our study started with a systematic literature review, in which we sorted through relevant conferences and identified potential needs and requirements sources in the literature. We then extracted, analyzed and organized these into requirements fitting predefined relevance and inclusion metrics. The resulting set of requirements was then validated and consolidated using expert feedback via a survey. This was rounded out by a quantitative analysis of the survey results.

The resulting set of requirements is useful both as a metric for RE tool capabilities and as a guideline for the development of model-based RE tools, including our own research prototype. While many of the analysis results empirically reinforce existing preconceptions, some results challenge established wisdom, e.g., the apparent lack of interest in highly automated validation and verification of requirements. This may necessitate a change in the way academia presents the practical usability and benefits of MBRE.

I. INTRODUCTION

Model-based requirement engineering is an approach to requirements engineering in which the requirements and the related business and development information, which is to be documented and specified, is collected, organized and structured, not only by use of natural language but also with the help of models described by formal, semi-formal or informal described modeling languages. There are different types of models, e.g., conceptual models, structuring the different kind of artifacts elaborated during requirements engineering, as well as model classes representing different types of artifacts, such as domain and goal models.

63% of respondents to a representative survey of software engineering practices in the medical devices industry stated RE activities as the origin of most software related problems. "Complexity presents a critical issue in the engineering of large, software-intensive systems. [...] Such complexity can only be mastered by using appropriately chosen models, [...] and providing support through theories and tools." [6]. "Experienced practitioners in requirements engineering in the embedded systems (ES) domain agree that models often or always help them understand complex requirements more easily and express a strong wish for using models more intensively in RE. One major obstacle for using models more intensively in RE is the lack of appropriate tool support" [7].

When speaking of model-based development, UML often comes to mind. UML provides an interesting basis for model-based development and is fairly widespread (more than a third of our survey participants use UML), yet it has problems and deficiencies particularly concerning use case modeling and system decomposition, as described in detail here [8]. Notwithstanding its claim to universality, UML is furthermore a semi-formal language based on loosely fitting aspect models with weak semantic links. Overcoming many of these general problems calls for the heavy use of customized UML-extensions and stereotypes, which increases the dependency between the understandability of the expressed constructs and one's understanding of the (re-)used language elements used to represent them.

In the course of developing our own MBRE academic research tool, we searched for the core set of language- and method-independent requirements such a tool would have to fulfill. Our search could not identify such a concise list of requirements in the literature. Many sources cite problems and challenges for MBRE in embedded systems, a few derive needs from these challenges and fewer still gathered these needs systematically. Recently, there are studies collecting and analyzing the needs of industry (e.g., [7]). No sources we found translate these needs into a concrete prioritized requirements list that we could use, as explained in detail in II. The consolidation of these different works is as yet lacking in the RE community, but given the relatively high feedback rate we received to our survey, it is a practically pertinent question, which we decided to answer ourselves.

The results of our survey serve as an implementation guideline for our own research prototype MIRA (model-based integrated requirements analysis) [9]. The goal of MIRA is to provide requirements engineering capability for projects modeled in our research software development tool AUTOFOCUS 3 [10], providing seamless model-based development for
embedded systems. The most relevant requirements identified in this paper have already been integrated into MIRA.

A. Main research question

What is the core set of MbRE specific requirements that a practically usable tool for embedded systems would have to fulfill?

B. Approach

Our solution comprises a study starting with a systematic literature review; identifying and sorting relevant sources. We then proceeded to extract, analyze and organize any clear requirements which fit predefined relevance and inclusion metrics. The relevance of this requirement collection was then validated by expert feedback via a questionnaire. As a bonus the experts were asked to choose five requirements they would prioritize highest. Moreover, the experts were asked to supply requirements not included in our list but very important in their opinions. Finally, we performed a quantitative and qualitative analysis.

C. Findings and consequences

While the analyses lend proof to many preconceived notions, e.g., higher demands on tool functionality by experts working on safety-critical products, it also resulted in some surprising discrepancies. For example, there is an apparent lack of interest in highly automated requirements validation and verification, with only 32% of the participants rating it as must-have and even fewer including it in their top five list.

Beyond its immediate merit for evaluating the importance of each of the requirements under study, the resulting systematic collection of consolidated and prioritized requirements for MbRE tool support could serve as a specification for the development of such tools as well as a comparison and evaluation index for existing tools. The findings suggest a divergence in the perceived maturity of advanced MbRE features between industry and academia. It may be beneficial to rethink the way academia presents the practical usability and benefits of MbRE to industry experts. Additionally, the perceived low importance and priority of advanced tool requirements, as explained in VIII signifies that these are not yet fulfilled or handled well by the existing tools.

D. Paper structure

Chapter II gives an overview of related works. In chapter III we present our approach. Chapter IV details how the literature review was carried out. The definition of requirements is presented in chapter V. The subsequent expert questionnaire is presented in chapter VI and its results in chapter VII. In chapter VIII we take an analytical look at the results and present our findings. Chapter IX gives the customary summary and outlook.

II. RELATED WORK

Our focus is the core set of requirements specific to model-based development that a MbRE tool for practical use in embedded systems would have to satisfy, without being bound to a certain method.

Several academic studies exist enumerating the many challenges and problems facing model-based software development in general, as well as MbRE, and providing visions for improvement. [11] discusses challenges in automotive software engineering, pointing out the importance of tools in model-based development, also in RE. [12] discusses challenges and research directions in RE, mentioning modeling activities without any focus on tools and tooling requirements. [13] points out challenges in safety for software intensive systems like medical devices, which were extracted and transformed into requirements. [14] gives a research roadmap on model-driven development of complex software, but without focus on RE.

The RE tools standard ISO/IEC TR 24766:2009 [15] gives general tooling requirements for RE without addressing the challenges of model-based development. Standards for embedded systems, such as the ISO 26262 automotive safety standard [16], contain domain-bound, implicit and disjoint requirements for RE activities and development tools viewed from a very narrow context-driven perspective (in this case safety). [17] specifies the language requirements for a user requirements notation in the field of telecommunications. In some contexts, such as requirements on tracing and refinement, language requirements overlap with tool requirements. We used this source to reinforce common requirements derived from more general sources that handle embedded systems as a whole.

Context-specific studies exist which focus on a given tool, like [18] or pertain to a specific method such as [19], [20]. Where possible, we extracted generally useful requirements within our tool- and method-independent focus. A deep analysis of these sources to extract generally useful requirements is outside the scope of this work. [21] used an approach similar to ours while focusing on the common requirements for a research prototype tooling platform.

Few studies, such as [22], which presents an evaluation framework for verification and validation, come close to our work in their corresponding focus areas. We extracted several requirements from this source. [7] collects the needs and expectations that the industry has regarding RE methods and served as one of the starting points for our work. It is not specifically focused on requirements for tools, but was an important source for our requirements set.

We could find no source that holistically presents, discusses or evaluates a collection of requirements for a practical MbRE tool of embedded systems. We found very few sources that target almost all focus points (e.g., MbRE tool requirements but not embedded-system specific) and as such we also included those sources that were pertinent to but not specifically targeting all our focus areas.
III. Approach

Figure 1 gives an overview of the chosen approach presented in this chapter.

A. Scope and Limitations

In the course of extending our research CASE tool [10] with RE capability, we needed to determine the core set of method-independent requirements a practically useful MbRE tool would have to fulfill. Our focus was on the added functionality provided by model-based approaches in general, especially those going beyond the capabilities of purely text-based (albeit structured) RE tools, e.g., the possibility to simulate executable requirements. Approaches based on the explicit support of concrete model types or modeling languages, which have been covered in [7], were not within the scope of our study. The use of model-based approaches inherently necessitates some tool features or capabilities, e.g., the capability to trace from textual requirements to models, so we also focused on these requirements. All requirements are elicited and evaluated in the context of embedded systems.

To avoid repetition and keep within the scope of our study, general requirements for software development tools, such as version or configuration management, were ignored. Also, requirements for model-based development tools, which were not specific to RE, such as model transformation capability, were not considered. Finally requirements for RE tools that are not specifically related to the use of models are not scope of this work. Hence, general requirements such as tracing and refinement out of MbRE-specific context are not listed as such.

B. Context

The context of MbRE in embedded systems necessitates special attention to the increased interdisciplinarity and complexity as well as domain-specific knowledge in such systems. The embedded systems domain has many fairly rigid standards which bring about their own requirements, e.g., the many safety standards (IEC61508, CENELEC, DO178B, etc.). We chose to evaluate the ISO 26262 Functional Safety in Automotive Systems standard [16] because it is a representative instance of such embedded systems specific knowledge. Having been released in its final form in November 2011, it represents the newest and most relevant instance of developments in embedded systems standardization.

C. Elicitation

In order to determine the relevant requirements for a MbRE tool for embedded systems, we wished to cover both the prespective of sources of research and knowledge of practical state-of-the-art. We discussed several alternatives and narrowed them down to

1) a systematic literature review,
2) by conducting interviews with researchers,
3) examine existing tools, or
4) ask experts from industry in regards to practical needs.

An actual in-depth examination of the multitude of existing tools would only yield results as to the status quo from the tool provider perspective. Given our focus, we opted for a triangulation approach [23] starting with a systematic literature review on scientific articles in search of MbRE challenges, needs and requirements. The ISO 26262 safety standard is then analyzed and used to both consolidate the results in an intermediate step and as a source for additional context-specific requirements. A feedback survey with industry experts rounds out the process.

D. Validation

The collected requirements were validated with regards to the aspects correctness, completeness, relevance and priority. Correctness and completeness were partially validated by comparing the requirement set collected in the first step (literature review) with the results of the analysis of the ISO 26262 standard. All aspects were then further validated using the expert feedback via a survey, coupled with our own analyses to achieve a concise list of MbRE tool requirements.

E. Threats to Validity

The main threats to validity can be internal (design of the study) or external (representativeness of the results) as well as threats to the construct (choice of measured and independent variables and whether they represent subject of study), conceptual (unnecessary duplication or theoretical isolation of the subject), and statistical conclusion validity (inappropriate use, inadequate application of statistical techniques, mistaking trivial effects for the support of the research hypothesis).

Construct validity
Due to our focus on evaluating the relevance of preselected requirements, we chose to limit our questions to the most obvious one: a rating of the requirements and prioritized refinement of these results by having the practitioners select their five "most relevant" requirements at the end of the survey. In order to make sure this limitation did not lead to an oversight we gave the participants the chance to add any requirements they thought we had missed.

Internal validity

The respondents may have a different understanding of the terms we used in the questions. We therefore tried to use a homogenous set of commonly understood terms throughout the entire survey. Furthermore, despite our best efforts to formulate the requirements in a simple and concise manner, we identified the risk of false data in the survey due to misunderstandings. To mitigate both these risks we added the possibility to rate requirements as unclear.

The results might be influenced by the layout and order of questions in the questionnaire. To mitigate this threat, we provided a field for further comments.

We also intentionally left out the sources of the requirements and any hints thereto in order to avoid an evaluation bias in the subsequent survey due to the participants’ trust or lack thereof in the requirement source.

External validity

Self-administered surveys are generally threatened by low response [24] rendering the results statistically insignificant. To preemptively counteract this we chose to target a respondent pool with high experience in requirements engineering and model-based software engineering in order to increase the quality of the survey results and our confidence in them, independently of the number of respondents. We did this by directly selecting known addressees with a high affinity to our target topics. We decided against using the address lists of large industry and research associations because we would be unable to evaluate the population response to our study or have any control over the relative quality of responses. The long experience of the globally active participants with multiple industry sectors and establishment sizes and types being represented as well as the respondents’ high interest give the results a high quality, as discussed in detail later in VII. We believe this balances out the small (in absolute terms) number of respondents and renders the results representative within the scope of our study.

Most of our respondents are located in Germany, which may threaten the transferability of our results to other countries. However, many of these German respondents work in globally active companies with experience in internationally distributed projects. Along with the respondents from other countries we believe this ameliorates the impact of this threat.

Finally, the various company sizes of the respondents may affect the results, as the RE needs and activities of small and medium enterprises are different from those of large companies with well-established processes. Our study was not setup to highlight these differences and it was not within the planned scope of this paper. Other studies exist, such as [25], which investigate this topic. We believe the input from both sides of the size spectrum makes our results more generally relevant.

IV. Literature review

A systematic literature review is a means of identifying, evaluating, and interpreting all the available research that is relevant to a particular research question, topic area, or phenomenon of interest [26]. As we had a fairly specific idea of what we were searching for, a systematic literature review as described in [26] and [27] was, per definition, a good starting point to answering our first research question (RQ).

RQ1: What are the core requirements of a MbRE tool for embedded systems?

The activities involved in a systematic review can be grouped in three phases: planning, conducting, and reporting. In the planning stage, the needs of the review are identified, the research questions are specified, and the review protocol is defined. In the conducting stage, the primary studies are selected, the quality assessment is defined, the data extraction and monitoring are performed, and the obtained data is synthesized. Finally, in the reporting stage, the dissemination mechanisms are specified, and the review report is presented [26]. The planning and initial parts of the conducting activities are described in the following subsections of this chapter. The description of the extracted data and our synthesis process along with the reporting activities are presented in the next chapter V.

A. Keywords

In our literature review we concentrated on the keywords "model-based", "requirements engineering", "software engineering" or "embedded systems" and synonyms like "software development" in order to find the relevant literature for requirements on a model-based requirements engineering tool for embedded systems.

B. Search sources

A standard search in IEEE Explore and Google Scholar did not lead to many fitting results. We used the few relevant returns along with a rough keyword search in Google (G) to identify relevant venues for our literature study. Based on these results and adding other conferences and journals known to us for their specific context, we sorted possible sources divided into conferences (C), journals (J), a symposium (S), workshops (W) and a standard (ISO), listed in table I. We kept our focus on relatively newer contributions.

C. Inclusion criteria

We did a manual analysis on the title and abstract of the papers in order to select a starting batch of papers for a deeper analysis. After a few iterations we settled on the following inclusion rules:

- the publication date is generally between 2007 and 2011
- the paper contains the key words ("model-based" AND ("requirements engineering" OR "embedded systems")) or synonyms thereof
TABLE I
LITERATURE REVIEW SOURCES

<table>
<thead>
<tr>
<th>Type</th>
<th>name</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Google key word search</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>C</td>
<td>IEEE International Requirements Engineering Conference (RE)</td>
<td>2006</td>
<td>2011</td>
</tr>
<tr>
<td>C</td>
<td>International Conference on Advanced Information Systems Engineering (CAiSE)</td>
<td>2009</td>
<td>2011</td>
</tr>
<tr>
<td>C</td>
<td>International Conference on Software Engineering (ICSE)</td>
<td>2006</td>
<td>2011</td>
</tr>
<tr>
<td>C</td>
<td>IEEE Signature Conference on Computers, Software, and Applications (COMPSAC)</td>
<td>2007</td>
<td>2010</td>
</tr>
<tr>
<td>C</td>
<td>Embedded Real Time Software and Systems (ERTS²)</td>
<td>2004</td>
<td>2010</td>
</tr>
<tr>
<td>J</td>
<td>Requirements Engineering</td>
<td>2006</td>
<td>2011</td>
</tr>
<tr>
<td>J</td>
<td>Transactions on Software Engineering and Methodology</td>
<td>2007</td>
<td>2011</td>
</tr>
<tr>
<td>J</td>
<td>Journal of Systems and Software</td>
<td>2009</td>
<td>2011</td>
</tr>
<tr>
<td>S</td>
<td>The Future of Software Engineering (FOSE)</td>
<td>2007</td>
<td>2007</td>
</tr>
<tr>
<td>W</td>
<td>Dagstuhl-Workshop: Model-Based Development of Embedded Systems</td>
<td>2006</td>
<td>2011</td>
</tr>
<tr>
<td>W</td>
<td>Model-based Methodologies for Pervasive and Embedded Software</td>
<td>2005</td>
<td>2010</td>
</tr>
<tr>
<td>ISO</td>
<td>ISO 26262 Road vehicles – Functional safety</td>
<td>2011</td>
<td>–</td>
</tr>
</tbody>
</table>

- the paper does not explicitly exclude embedded systems
- the paper describes problems, needs or tool requirements in requirements engineering

D. Exclusion criteria

We did not consider papers focused on requirements specific to certain methods or to a concrete research tool, because we wanted to create a general method-independent set. The analysis of these papers for generally valid requirements was carried out where possible.

E. Selection and sorting

We had originally planned to only rate sources which simultaneously covered all keywords as highly relevant. The first resultant batch of papers, albeit very large, yielded very few sources matching all of our criteria. We thus had to relax our initial relevance criteria, by redefining highly relevant papers as those which handle the tool needs and requirements of MbRE in general, without limitation to embedded systems applications. To cover the embedded systems aspect we included also all papers covering tool needs and requirements for model-based software or systems engineering in embedded systems, without special emphasis on requirements engineering.

This batch was then classified using the apparent focus of each source (as described in the abstract or by quick overview of the content) into four categories.

- **Tool needs**: All papers containing problems (incl. open questions), that could be solved by a tool, needs from practitioners for a tool or tool requirements
- **Tools**: Papers describing existing tools, their uses and methodologies, especially research tools
- **RE method**: Papers describing a requirements engineering method
- **Misc.**: Papers containing pertinent information, which does not fit into the other categories

F. Literature review results

We analyzed six conferences, six journals, one symposium and two workshops covering a period of around six years with a total of around 3300 papers. Through the manual analysis on the titles or abstract of these 3300 papers we identified 130 papers, whose subject matter warranted a deeper analysis. Using the prioritization criteria discussed in IV-E we narrowed these down to 28 papers pertaining to tool needs, which we classified as highly relevant to the topics discussed in this paper. A list containing these 28 papers can be found online at [28].

V. REQUIREMENTS DEFINITION

A. Problems, needs and requirements extraction

The selected papers from the literature review were analyzed in depth with the purpose of eliciting concrete problem statements, needs and, if available, requirements. Using the scope defined in chapter III-A, we extracted 46 statements from 11 different sources, [29], [30], [31], [32], [13], [22], [33], [34], [7] and [35] as well as the ISO 26262 [16].

B. Requirements derivation and statement

Most of the statements we collected had a form more apt for problem and need description and we thus had to refine and hone them into requirement statements. Where such a clearer statement was not possible, the original statement was omitted. Due to their disparate sources the collected statements did not use homogenous wording. We thus proceeded to harmonize the statements; using common terminology, consolidating similar requirements, deleting duplicate ones and finally reformulating them into concise statements, where necessary and possible. This last step was taken to prevent longer statements causing disinterest in the survey participants. This led to the derivation of 23 core MbRE tool requirements for embedded systems, which we then grouped into seven requirement categories. The categories are called 1) general requirements, 2) requirements documentation, 3) abstraction layer, 4) tracing, 5) tool integration and customization, 6) validation and verification, and 7) further functionality. The categorized requirements are listed in figure 2.
VI. Survey

In order to consolidate and validate the list of requirements collected in the previous step, we devised the following research questions for experts in the field.

RQ2: Are the collected requirements important? to check the correctness and relevance of each single requirement.

RQ3: How would you prioritize them? to get an overall implementation prioritization among the requirements, as well as a finer relevance rating.

RQ4: Which requirements are missing? to validate completeness.

A. Design of the questionnaire

We set up our survey in the form of an online questionnaire, based on the guidelines described in [24], consisting of two blocks participant information and tool requirements.

Participant information: to evaluate the participants’ experience as well as their establishment and sectors. Their working experience and level of involvement with RE were mandatory questions. The following answers were possible: that RE is one’s main task, that the respondent is involved, but not as a main task or that it is part of everyday work, but not directly involved. We initially planned no involvement, but then removed this option as it would introduce unnecessary bias into the results, especially since no one working in software development for embedded systems is absolutely uninvolved with requirements engineering. To focus on our target group we excluded responses from participants for whom requirements engineering is part of everyday work, but they are not directly involved as well as any from the Applied Research / Education sectors. Respondents were also asked to state which models and modeling languages they use in requirements engineering, as well as whether they are involved in safety-critical product development.

Tool requirements: this block contains all requirements and an ordinal scale [36] to evaluate them. The scale is rated with the following, self-explanatory options, as an answer to our second research question RQ2:

- unclear requirement
- not relevant
- nice to have, but not necessary, hereafter shortly referenced as nice to have
- important, but could live without it, hereafter shortly referenced as important
- must-have

To answer RQ3, the respondents were asked to select the five highest priority requirements in their opinion. Finally we asked for additional requirements and comments to cover RQ4.

Although we use a questionnaire instead of interviews, the guidelines provided by [23] give a solid reference for planning and analysis. For the evaluation of the importance and priority of the requirements, we use the cross-case analysis and compare the results of the group of participants developing safety-critical products with non-safety-critical products and
set them in relation to all answers.

The additional requirements provided by the respondents were sorted based on content into three groups. The first group fits within our scope of MbRE requirements. The second group comprises requirements on the models themselves (e.g., use of a certain modeling language). Finally the third group comprises all other requirements, e.g., non-functional requirements like pricing or requirements on the workflow. We focused on the first two groups, extracting and consolidating relevant requirements.

B. Selection of participants' pool

Our target group for the survey is experts from industry, preferably with high experience in requirements engineering and model-based software engineering. The population is large enough to warrant settling for a sample for our study. We invited around 100 international experts in software and systems engineering from our contact lists. We chose participants matching our target criteria, with a high affinity to requirements engineering and model-based software engineering to increase the quality of the survey results and our confidence in them independently of the number of respondents. Participants could leave their email in order to get the results of the study.

VII. Expert Questionnaire Results

A. Evaluation of the replies

We received 29 filled replies within the scheduled month. After filtering the results according to the participant exclusion criteria explained in VI-A, 22 responses fulfilled the desired criteria regarding experience in RE and industry relevance. We focused all subsequent analyses on these 22 replies.

The quality of the received replies was very high in terms of completeness, with almost all participants filling out the optional fields as well.

All 22 participants gave information about their working experience in years. The average working experience in requirements engineering is 9.4 years, spreading from one to 25 years, the general working experience is 17 years, spreading from five to 29 years. The participants come from 5 different countries, with 16 participants coming from Germany. They work in seven different sectors, namely Aeronautics, Aerospace, Automotive, Consulting, Telecommunications, Systems Engineering and Information Technology and at least 10 different companies. 12 persons work on safety-critical products, 9 persons on non-safety-critical products, one person gave no answer. The company size varies between 1 and more than 250,000 employees, with 10 participants working in small and medium sized enterprises with up to 250 employees and 10 persons working in large companies with at least 1,000 employees. Two participants gave no detailed information on the company size.

B. Results

1) Modeling languages used by participants: 19 participants gave a list of different models and modeling languages they use. Most common are SysML mentioned by 8 persons, UML with 7, MATLAB/Simulink with 6 and EAST-ADL with 4 persons. The participants gave a wide field of other languages like BPMN, Event-B or specific DSLs, every language only stated by one person.

2) Research question 2: Are the collected requirements important? Each of the collected requirement is a must-have for at least 23% of the participants, which is less than expected. On the other hand, each requirement is considered as highly relevant; at least 50% of the participants rated each and every one of them as important or must-have. This percentage is even higher in the safety-critical products group, with 59% of the participants considering all of the requirements at least important.

The conclusion can be drawn that all of these requirements have to be considered when developing a MbRE tool.

As described in chapter III-E, the brief description of the requirements entails the risk that requirements are misinterpreted by the practitioners. This occurred in a total of 12 instances, each by one to three participants. Due to the low occurrence ratio, we do not believe this to have significantly influenced our results.

3) Research question 3: How would you prioritize them? The two requirements with the highest priority in the general evaluation are the support of different representation forms of requirements: informal (flowing text description), semi-formal (UML diagrams), formal (mathematics-based) and the support for document generation. Each is included into the high priority list by 8 participants. An overview on the requirements with the highest prioritization is given in figure 3. 3 participants gave no prioritization of requirements.

The safety-critical development experts gave less priority to document generation, but instead emphasized the importance of support for different representation forms as well as for different, well-defined system abstraction layers.

For the non-safety-related group the main requirement consists in support for the integration of tools and their artifacts, (e.g., domain specific tools integrated by import/export interfaces), followed by document generation.

Figure 4 shows the percentage distribution of the answers, classified by their prioritization and indicating a clear and strong correlation between importance and prioritization, in which almost 40% of must-have requirements were selected in the priority list while only 6% of the important requirements were prioritized. These two categories alone cover 77% of all requirements. The remaining categories, accounting combined for 23% of requirements, received negligible priority votes.

4) Research question 4: Which requirements are missing? One question in the questionnaire regards additional requirements in order to identify missing requirements. 10 participants gave 27 additional requirements. Six of them are non-functional requirements like usability, pricing, etc. One recommends supporting certain languages, two are requirements regarding the workflow and another two are on text processing. These requirements were therefore omitted from the list as they are not related to the scope of our study. The remaining 16 requirements were consolidated down to 11. The
### Requirements and Models

**Support different representation forms of requirements: informal, semi-formal like UML, formal (mathematic-based)**
- # of participants: 8

**Support document-generation**
- # of participants: 8

**Support the representation of non-functional requirements (safety, security, reliability, maintainability, etc.)**
- # of participants: 7

**High level of automation in creating and maintaining traceability links within requirements**
- # of participants: 7

**High level of automation in creating and maintaining traceability links between RE and design**
- # of participants: 5

**Support for the integration of tools and their artifacts, e.g., domain specific tools integrated by import/export interfaces**
- # of participants: 5

**Support for different, well-defined system abstraction layers (product, system function and component views)**
- # of participants: 5

### Literature Review

1) **The topic in literature:** The first finding of our literature review is an internal one: The topic is new; there are no sources that present, discuss or evaluate a holistic collection of requirements for a MbRE tool for embedded systems.

2) **Interpretation of the term model-based in requirements engineering:** In the papers we analyzed, the term model-based or model is often used without a clear statement or definition of what is meant by this term, with no apparent consensus.

### Survey

**ID**

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
</tr>
<tr>
<td>A2</td>
</tr>
<tr>
<td>A3</td>
</tr>
<tr>
<td>A4</td>
</tr>
<tr>
<td>A5</td>
</tr>
<tr>
<td>A6</td>
</tr>
<tr>
<td>A7</td>
</tr>
<tr>
<td>A8</td>
</tr>
<tr>
<td>B1</td>
</tr>
<tr>
<td>B2</td>
</tr>
<tr>
<td>B3</td>
</tr>
</tbody>
</table>

**Fig. 3. Requirements with the highest prioritization**

**Fig. 4. Percentage distribution of the importance of requirements, classified by prioritization**

**Fig. 5. Additional requirements**

The additional requirements are summarized in figure 5. There is a high interest in requirements regarding the standardization of requirements and models and their exchangeability, mentioned by four participants and consolidated into one requirement. Three requirements deal with tracing, again indicating this topic’s general importance.

The first eight requirements are the most relevant to our work. Of particularly noteworthy importance to the practitioners were requirements for the supplier / system-integrator interface (req. B1-B3). These requirements are abundantly mentioned in existing literature and are even defined in existing safety standards such as the ISO 26262 [15], but are not specific to model-based development. We decided to include these remaining three into the list of additional requirements, despite them being outside our scope, because of their perceived importance in the eyes of the practitioners participating in the survey.

### Discussion of the Findings

#### A. Literature review

1) **The topic in literature:** The first finding of our literature review is an internal one: The topic is new; there are no sources that present, discuss or evaluate a holistic collection of requirements for a MbRE tool for embedded systems.

2) **Interpretation of the term model-based in requirements engineering:** In the papers we analyzed, the term model-based or model is often used without a clear statement or definition of what is meant by this term, with no apparent consensus.

#### B. Survey
1) Relevance: Given that self-completion surveys generally suffer from low responses especially when coinciding with holiday seasons [24], we found our high rate of replies within one month with around 29%, despite the survey going out in late December, very interesting. The high percentage of replies could have resulted from our direct contact with the experts, but 82% of the participants explicitly left their email in order to get the results of our questionnaire. The participants’ strong involvement is also underlined by the high rate of completion of optional fields, especially the additional requirements. This increased interest indicates the topic is practically relevant.

2) Views of safety vs. non-safety practitioners: General practitioners focus on basic features of a MbRE tool in terms of documentation (req. 2b, 6c), automated tracing (req. 4a, 4b), with a clear preference on a simple integration into the tool chain (req. 5a) and the support of non-functional requirements (req. 1a). On the other hand, advanced features, like formal and semi-formal representation (req. 2a), or simulation of requirements (req. 6d), received relatively low importance and priority votes, as discussed further below.

SE Practitioners working on safety-critical products have a higher demand on tool functionality. This is underlined by the overall average importance of these requirements (rated as must-have and important) being significantly higher in this group. A deeper analysis of the individual ratings yields an even stronger emphasis and priority on more advanced features with many of these requirements receiving importance ratings over 90%. Different representation forms for requirements (informal, semi-formal like UML, formal) (req. 2a) have a very high priority and importance. High priority is also given to the support for different, well-defined system abstraction layers (product, system function and component views) (req. 3a). Experts from this group rate a deep integration of the MbRE tool in an existing development process with direct support for interactions between activities and consistency checks (req. 5b) more important than experts from the other group. Also, support for variant management (req. 7c) and software criticality levels (7b) play an important role.

An interesting finding is the apparent growing awareness of the importance of safety-critical activities among general SE practitioners and integrating them into RE, indicated by the non-safety-related practitioners group considering the representation and propagation of software criticality levels (7b) a must-have.

3) Perception of advantages of model-based requirements engineering: One very interesting finding is the comparably low importance some of the big advantages of model-based requirements engineering received. Among the promising features that were rated comparably low: Supporting a high level of automation of the validation and verification of requirements (req. 6a) and simulation of (executable) requirements (req. 6d).

The findings indicate a persistant misconception by general SE practitioners of the maturity of advanced features, e.g., formal methods, and their applicability in practical industry projects. Given their benefit, we expected these requirements to be much more important. It may be beneficial to rethink the way academia presents the practical usability and benefits of MbRE to industry experts. Additionally, the perceived low importance and priority of advanced requirements signifies that these are not yet fulfilled or handled well by the existing tools. These questions have to be investigated further.

IX. Summary and Outlook

A. Summary

Our contribution is a systematic collection of consolidated and prioritized requirements for an adequate model-based tool support together with rationales and sources from literature and standards.

Using a systematic literature review we identified an initial set of requirements for a model-based requirements engineering tool for embedded systems. This set was evaluated and extended by seasoned experts from industry with broad experience in requirements engineering and model-based systems and software engineering. This led to a validated and enriched core set of requirements.

While experts working on non-safety-critical products focus on basic functionality like documentation and automated tracing in their prioritization, the requirements engineers of safety-critical products have a higher demand on the functionality of a MbRE tool as shown by the higher importance rating of nearly all corresponding requirements. For example, the support for different, well-defined system abstraction layers (product, system function and component views) is much more important in a MbRE tool for the development of safety-critical products.

The analyses lend proof to many preconceived notions, e.g., higher demands on tool functionality by experts working on safety-critical products, as well as established guidelines, e.g., the requirement for clear interfacing between suppliers and system integrators. They do however also shed light on some surprising discrepancies. For example, there is an apparent lack of interest in highly automated requirements validation and verification, with only 32% of the participants rating it as must-have and even fewer including it in their top five list of prioritized requirements.

The goal of our evaluation is to show trends. A more accurate rating within these requirements is not possible, especially given the relatively small number of participants. The high rate of response, along with the participants’ strong involvement, indicates the relevance of the topic for industry. This warrants further work extending this collection of core requirements.

B. Outlook

This study serves as a starting point for further discussions and refinement.

The scope of the requirements collection could be further expanded and more widely consolidated in a larger study. The indicated relevance means there would be a high interest in industry to spend time and effort on this topic.
The consolidated results of this analysis serve as a specification handbook for the development of our own MbRE research prototype MIRA and will as well be used to define a metric which can be used to analyze and compare existing MbRE tools or serve as a guideline to implement new tools.

REFERENCES


