

# Model-based Development Tools for Embedded Systems in the Industry

## – Results from an Empirical Investigation –

Markus Herrmannsdoerfer, Thomas Kofler, Stefano Merenda, Daniel Ratiu, Judith Thyssen  
Institut für Informatik, Technische Universität München, 85748 Garching, Germany  
{herrmama, koflert, merenda, ratiu, thyssen}@in.tum.de

**Abstract:** Model-based development (MBD) is adopted in practice in different degrees. A prerequisite for building a new generation of tools for model-based development of embedded systems is to gain an overview over their current use and the needs of the industry. In this paper, we present the results of an empirical study—that we conducted in the context of the SPES2020 research project—about the use of model-based development tools in industrial projects. We investigated the following general research questions by collecting empirical data from several industrial project partners: (1) What is the status quo of the tools currently used in industry? and (2) What are the requirements for the next generation of industrial tools? The result of our study shows that there is a considerable heterogeneity in both the development tools and in the manner in which model-based development is perceived in different industrial branches (e. g., automation, avionics, automotive). Furthermore, we identified that the most important need for industrialization of model-based development in the future is a higher integration of different tools and methods.

## 1 Introduction

Model-based development promises to raise the abstraction level at which software development currently is performed and thereby to increase the quality and automation and to lower development costs. In the purest way, model-based development implies the pervasive use of adequate models in every development step and as main development means. In practice however, model-based development is adopted in different degrees. There is a wide spectrum under which model-based development is understood and applied in current industrial projects.

A prerequisite for a new generation of tools for model-based development of embedded systems is to have an overview over the current situation and the needs of the industry. In the context of the research project “Software Platform Embedded Systems” (SPES2020), we tackled this problem by performing a questionnaire-based empirical study. The main goal of this questionnaire was to get an overview over the situation and the needs of model-based development in industry today. In this questionnaire, we concentrate on questions regarding technologies and tools used to develop embedded systems. Although these questions focus on tooling issues, the results also reflect the methods used today. We conducted the questionnaire to investigate the following general research questions:

1. **What is the status quo of the tools currently used in industry?** From the tools currently used in industry, we aim at drawing conclusions about the methods used for the model-based development of embedded systems.
2. **What are the requirements for a next generation of industrial tools?** We aim at deriving requirements for a next generation of industrial tools.

According to our results, the most important issues today are that tools and technologies are considerably heterogeneous, that model-based development is not yet used to its full extent, and that existing tools are not yet fully appropriate for efficient model-based development. The most important need is the integration of existing tools for seamless model-based development. These results provide a valuable input for current and future academic research projects to ensure that the most important needs of the industry are taken into account.

The current paper is based on a deliverable written in the context of the SPES2020 research project [HM09]. It is part of a wider research area at the Chair for Software & Systems Engineering from Technische Universität München about tool architectures for seamless model-based development of embedded systems [BFH<sup>+</sup>10]. Our longer term goal is to take the status quo and requirements into account for the tool architecture which should be developed as part of the SPES2020 project.

**Outline.** In Section 2, we briefly present the design of our study. In Section 3, we present the results obtained after evaluating and interpreting the answers. We focus here mainly on qualitative results and only exemplary present several quantitative data. In Section 4, we discuss threats to the results' validity, before we conclude in Section 5.

## 2 The Empirical Study

In this section, we provide an overview of the design of our study. To answer the research questions, we developed a questionnaire which was distributed to the industrial partners of the SPES2020 project.

**Subjects of the study.** We performed our study by distributing the questionnaire to all industrial partners of the project SPES2020. They originate from a wide variety of domains in the field of embedded systems ranging from automotive over avionics or industrial automation to energy. Furthermore, some of the partners play the role of suppliers, while others play the role of system integrators.

**Design of the questionnaire.** The questions from our questionnaire can be divided into the following categories:

- *Context.* The questionnaire should be filled out from the perspective of a certain project in which the respondent works. The questions in this category ask for some

information about the project and the respondent's role in the project.

- *Target platform.* This category contains questions about the target platform onto which the developed embedded system is deployed. We have identified the four aspects which characterize the target platform (i. e., programming language, operating system, bus system, and middleware).
- *Overview of today's tools.* This category contains questions about the status quo of tools currently used in industry. The questions are subdivided into the following two classes:
  - *Nature of tools.* This category concentrates on questions about the tools currently used in industry. It is further subdivided into the categories *vertical tools* and *horizontal tools*. *Vertical tools* focus on a special development phase – we were interested in tools used in requirements management, modeling, and quality assurance. *Horizontal tools* are necessary for all development phases – we were interested in cross-cutting tools like those that belong to the infrastructure, tools for configuration management, change management or process support.
  - *Availability of tools.* This category concentrates on the questions whether the needed tools are available on the market and what the industry does in order to achieve the wanted tool functionality.
- *Deficits of today's tools.* This category contains questions about problems with the current tools and about requirements for future tools. We concentrate on the missing features and their impact, as well as the current and desired integration of the used tools.

**Types of questions.** In the questionnaire, we asked three different types of questions: (1) *One possible answer.* In these questions, we provided a number of answers from which the respondents should choose one. For missing answers, we added the additional category *no answer*. (2) *Multiple possible answers.* In these questions, we provided a number of answers from which the respondents could choose multiple ones. Furthermore, we included a category *other* through which the respondents could name additional answers. (3) *One possible answer for different aspects.* In these questions, the respondents should choose one answer for different predefined aspects.

**Execution.** To answer the research questions, we developed the questionnaire which consists of 25 questions in total. We built a first version of the questionnaire which was presented to a small number of project participants. Based on the feedback we got, we improved the questionnaire. The second version of the questionnaire was distributed via mailing list to all industrial partners. We altogether received 24 completely filled out questionnaires.

### 3 Results of the Empirical Study

In this section, we discuss the results of the questionnaire in terms of the research questions posed at the beginning of this paper. In Section 3.1, we discuss research question 1 – i. e., to what degree model-based development is adopted in industry today. In Section 3.2, we discuss research question 2 – i. e., which tooling deficiencies hamper the pervasive use of model-based development and what are the requirements for a next generation of industrial tools. Due to the limitation of space, we mainly present qualitative results. The detailed quantitative results of the questionnaire can be found in [HM09].

#### 3.1 What is the status quo of the tools currently used in industry?

Following our study, when looking at the current status of model-based development tools in industry, we make the following qualitative observations:

**Heterogeneity of target platform.** The target platform onto which the embedded system is deployed has a considerable heterogeneity. The target platform varies a lot from project to project in terms of both programming language, bus system, operating system, and middleware. It is important to note that this heterogeneity is not only between different companies, but also within a single company. As a consequence, the development tools are also affected by heterogeneity, as each of them is usually specialized for a certain target platform.

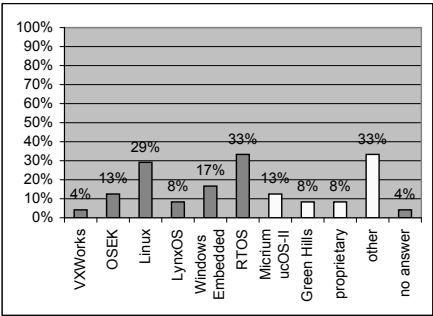


Figure 1: Which operating systems are running on the developed system? (Multiple possible answers.)

To characterize the target platform, we investigated the four aspects programming language, operating system, bus system, and middleware. Exemplary, Figure 1 shows the distribution of the answers to the question which operating systems are running on the developed system. The most widely used operating systems are real-time operating systems (RTOS) with 33%, Linux with 29%, and Windows Embedded with 17%. 33% of the respondents employ *other* operating systems like Embedded SW, MACOS (Thales), Microware OS-9 RTOS, Simatic, Simatic TDC, Sysgo (Pike OS), Timer Interrupt Driven

Proprietary Software Sequencer, TinyOS, Windows XP, and a simple scheduler. The extremely huge number of about 20 different operating systems clearly illustrates how heterogeneous are the platforms addressed by the development of embedded systems.

**Heterogeneity of tools.** There is also a big heterogeneity in the tools that are used for the model-based development of embedded systems. This holds for both vertical tools, i. e., tools which are specific to a certain development phase, and horizontal tools, i. e., tools which are necessary for all development phases, like version control systems or change management tools. Again, this heterogeneity is not only between different companies, but also within a single company. The heterogeneity of tools hampers their integration.

The results show that horizontal tools vary a lot in terms of the tool platform as well as for the activities of version and change management. Vertical tools vary a lot for the activities of requirements management, modeling, and quality assurance.

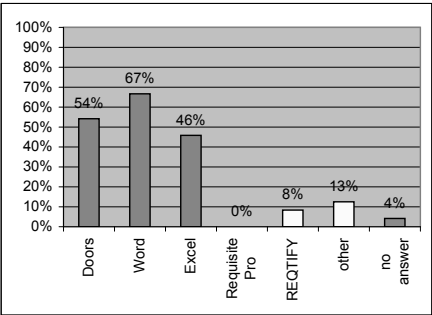


Figure 2: What tools are used for requirements elicitation and management? (Multiple possible answers.)

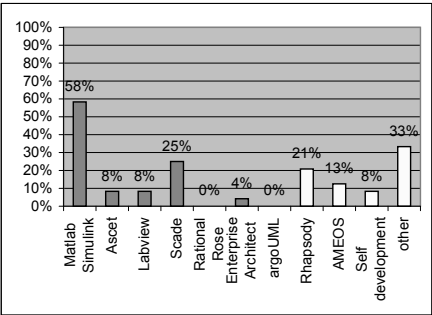


Figure 3: What modeling tools are used? (Multiple possible answers.)

Figure 2 shows the distribution of the answers over typical tools for requirements elicitation and management. Even though they are not tools tailored for requirements management, Microsoft Word and Microsoft Excel are still the most widespread tools with a share of 67% and 46%, respectively. However, Doors is used as a specific requirements management tool in more than 50% of the answers.

Figure 3 shows the share of typical modeling tools in the answers. With a usage rate of 58%, Matlab Simulink is the most widespread modeling tool—followed by SCADE with 25%. The UML and SysML tool Rhapsody is only on the third position, with a share of 21%. Besides Rhapsody, there are other UML tools used in industry like AMEOS and Enterprise Architect. 33% of the respondents named *other* modeling tools like MID Innovator, NX, PCS7 Engineering, radCase, SiGraph, Solid Edge, TAU, Teamwork, UML, X32 BlueRiver, and “not specified”. The high number of employed modeling tools shows how heterogeneous the tool landscape for modeling is.

**Weakly defined models.** Tools which are based on models without a precise meaning are still widely used in industry. For requirements management, the industry still relies a

lot on Microsoft Word and Excel which are general purpose and thereby not completely adequate for modeling requirements (cf. Figure 2). The situation is much better for system modeling with a high usage of modeling tools such as Matlab Simulink and Scade (cf. Figure 3). Models with a precise meaning can be better analyzed and the tools can offer a higher automation. Consequently, there is still much opportunity for improving the appropriateness of tools in order to work with semantically rich models and thereby to benefit from the promises of model-based development.

For process support, a lot of tooling is based on MS Project and Excel – with a share of 79% and 63%, respectively, as illustrated in Figure 4. However, both tools allow the developers only to define the process in an informal way and thus do not allow automation.

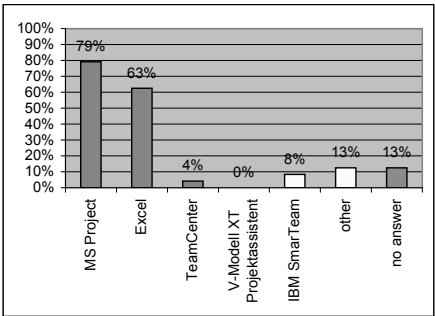


Figure 4: What tools do you use for process support? (Multiple possible answers.)

**Sporadic generation of code.** The tool chains used in practice do not yet provide the full benefits of model-based development. Even though the industry uses a lot of advanced modeling tools like Matlab Simulink and Scade, a lot of code is still handwritten instead of being generated from high-level models. As illustrated in Figure 5, only 16% of the respondents generate from 80% to 100% of the code. 34% generate from 40% to 70% of the code, and 42% generate only 0% to 30% of the code. An interesting point is, however, that there are already some companies that claim 100% code generation in certain projects.

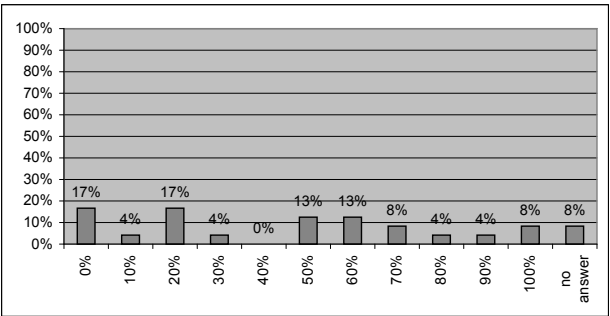


Figure 5: How large is the fraction of code generated from models? (One possible answer.)

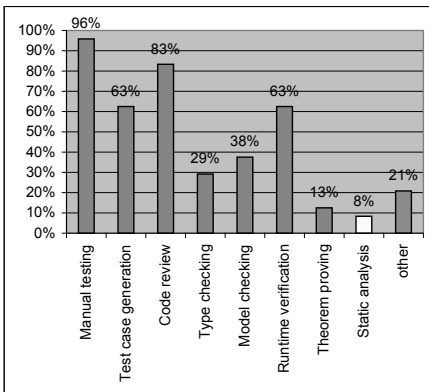


Figure 6: Which methods do you use for quality assurance? (Multiple possible answers.)

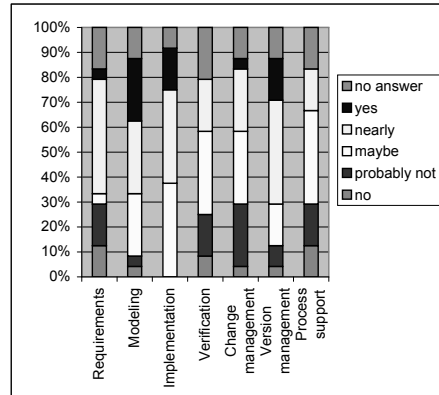


Figure 7: Do you think that you use the best tools available on the market? (One possible answer for each aspect.)

**Predominance of manual quality assurance.** For quality assurance, the industry still uses a lot of manual techniques like manual testing and code review. As illustrated in Figure 6, nearly every project performs manual tests and code reviews for quality assurance (96% and 83%, respectively). Automatic techniques for model-based quality assurance like model checking and theorem proving are not yet widely adopted in industry. This may be due to the fact that appropriate tools are not available on the market, or the respondents cannot use them due to other restrictions, like e. g., the missing interoperability with the current tool chain.

**Inability to use best-in-class tools.** A lot of respondents to the questionnaire are not able to use the best tools available on the market. More specifically, the results show that only 15% of the respondents think that they use the best tool (average over all process phases, cf. Figure 7).

A reason for this may be the high barriers to switch to another tool because of the high migration costs. For instance, migration effort is necessary for migrating existing data as well as for training developers to use the new tool. Sometimes, it may be even impossible to switch to another tool, as the new (and presumably better) tool may not integrate well within the existing tool chain.

**High rate of proprietary tools.** The industry makes use of a lot of proprietary tools (instead of commercial-off-the-shelf (COTS) tools). In particular, the results show us that proprietary tools are developed for all process phases. Although a huge number of development COTS tools are available, the market does not seem to provide solutions satisfying the requirements of individual projects. This may be also due to the missing customizability of existing tools to the individual requirements. In particular, most effort is spent on the integration of existing tools which is very important for seamless model-based development.

3.2 What are the requirements for a next generation of industrial tools?

Following the results of our questionnaire, when discussing the major needs of future development tools, we can make the following observations:

**Importance of tool integration.** A deep integration of the development tools is seen as the major issue. Most effort for proprietary tool development is spent on coupling of existing tools. This is consistent with the result that—with the share of 71%—tool integration is the most missing feature of today’s tools followed by process support with 42% (cf. Figure 8).

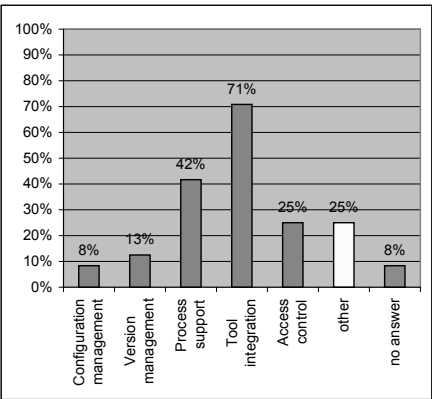


Figure 8: What features do you miss in the tools you use today? (Multiple possible answers.)

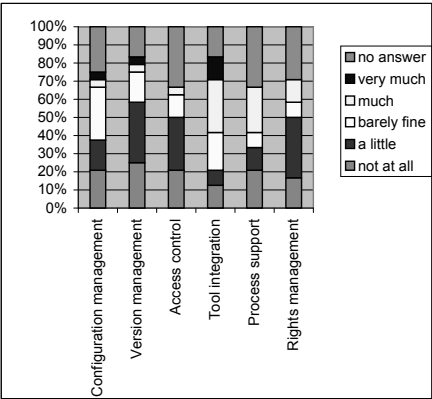


Figure 9: How far is your work restricted by the missing features? (One possible answer for each aspect.)

Moreover, the missing integration of the tools heavily restricts the engineers in their daily work. Figure 9 shows the distribution of the respondents’ rating with respect to the different activities in which tool support is required. The numbers show that the major restrictions are caused by the missing integration of tools: More than 40% of the respondents voted that the missing tool integration restricts their work much or even very much.

**Importance of process support.** Support of the development process is seen as an important feature which is not fully satisfied by current tools. Currently, the respondents mainly use Microsoft Project and Excel for process support (cf. Figure 4), but at the same time do not think that they use the best tools on the market (cf. Figure 7). However, the results show that they do not invest too much effort into the development of proprietary process support. This is probably due to the fact that good process support also requires tool integration which is also missing today. After tool integration, the respondents see process support as the most important missing feature (cf. Figure 8). The missing process support restricts a lot of respondents in their daily work (cf. Figure 9).

**Back end integration is more important than front end integration.** Integrating the back end is seen to be much more important than integrating the front end. In particular,



back end integration techniques like a central repository, tool couplers and consistency checks are considered as highly important, whereas integration of the user interface is not as important (cf. Figure 9). This seems to be the right prioritization, since without an integrated back end an integration of the front end cannot provide significant advantages.

## 4 Threats to Validity

We are aware that our results can be influenced by various threats to validity. They can be divided into threats to internal (design of the questionnaire) and external (representativeness of the results) validity.

**Internal validity.** The results might be influenced by the design we chose and the way we posed the questions for the questionnaire.

*Implicit assumptions in questions.* We make implicit assumptions about what is the optimal scenario of model-based development (e. g., full code generation and full tool integration are the optimal solutions, ...). However, these assumptions are in line with the paradigm and the promises of model-based development.

*Priming through predefined answers.* The selection of the possible answers may influence the result of the questionnaire. To mitigate this threat, we provided a field to specify *other* answers.

*Different understanding of terms.* The respondents may have a different understanding of the terms we used in the questions (e. g., “tool platform”, “tool integration”,...). To mitigate this threat, we tried to use commonly understood terms as far as possible.

**External validity.** The results might be influenced by the fact that we questioned only a subset of the possible developers of embedded systems and might thereby not be representative for model-based development of embedded systems in general.

*Number of respondents.* We received only 24 filled out questionnaires which poses a great threat to the results’ validity. However, the obtained answers come from all project sizes, different levels of project success, different domains where embedded systems are developed, and all process phases.

*Distribution of respondents.* The respondents were not equally distributed over all participating industry partners – taking their size into account. We received more filled out questionnaires from smaller companies, but fewer from bigger ones. However, we can still derive that the tools and technologies used for the development are quite heterogeneous today.

*Locality of respondents.* All the questionnaires were filled out by respondents located in Germany – threatening the transferability of our results to model-based development in other countries. However, we also had respondents working in companies that act globally as well as in globally distributed projects.

## 5 Conclusion

In this paper we presented the results of an empirical study to better assess the current situation of model-based development of embedded systems in the industry. According to our two research questions, we are able to come to the following conclusions:

Firstly, we found a considerable heterogeneity in the development tools as well as the target platform onto which the software is deployed. This heterogeneity makes it difficult to seamlessly integrate the tools and methods for model-based development. The results also show that model-based development is currently far from being used to its full extent in industry. Partially, only weakly defined models are used which makes it impossible to obtain a high automation and to employ advanced techniques for model-based quality assurance. Additionally, high migration costs from one tool to another hamper the usage of the best tools on the market.

Secondly, we wanted to identify the most important needs for model-based development in future. Our results show that the integration of tools and methods is considered by engineers as the most promising way to more efficient model-based development. The integration of the tools is not only important on the level of the models, but also on the level of the process. However, seamless integration on the higher levels (e. g., the process) requires integration on the lower levels (e. g., the models).

The success of model-based development of embedded systems in future depends on the deep integration of tools and methods. This includes the adaptability of tools and the flexibility of methods to support the overall process of developing embedded systems in an appropriate way.

Many tools used by our participants have quite some functionality in common, e. g., persistency, configuration management, and so on. A possible solution to integrate these tools is to build an integrated tool platform that factors out this common functionality [BFH<sup>+</sup>10]. Our long term goal is to provide the necessary concepts for building such an integrated tool platform. This integrated tool platform should be extensible with new modeling languages and should provide a mechanism to integrate these languages. To reduce the effort, modeling languages themselves should also be built using model-based development by defining explicit metamodels. Moreover, an integration of the modeling languages enables the tool platform to provide process support throughout model-based development.

## References

- [BFH<sup>+</sup>10] Manfred Broy, Martin Feilkas, Markus Herrmannsdoerfer, Stefano Merenda, and Daniel Ratiu. Seamless Model-based Development: from Isolated Tools to Integrated Model Engineering Environments. *Proceedings of the IEEE - Special Issue on Aerospace & Automotive*, to appear in 2010.
- [HM09] Markus Herrmannsdoerfer and Stefano Merenda. Result of the Tool Questionnaire. Technical Report TUM-I0929, Technische Universität München, 2009.