Peter Heidl
Jens Höfflinger
Harald Hönninger
Bastian Tenbergen

# 17

# Lessons Learned

*The purpose of this chapter is to offer a view on the lessons learned from developing and applying the SPES modeling framework. The lessons learned show that the SPES modeling framework is well-aligned with regard to the industry challenges. In addition, results show that the SPES modeling framework has the potential to provide a stable foundation for the further evolution of model-based approaches for embedded system development.*

K. Pohl et al. (eds.), *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*,
DOI 10.1007/978-3-642-34614-9_17, 

As proposed in Chapter 2, the engineering of modern embedded systems is becoming increasingly challenging due to a steady increase in demand for more innovations in shorter time to market as well as overall cost pressure. Current development approaches for the engineering of software-intensive embedded systems are ill-equipped to meet these high demands. Consequently, novel development paradigms are necessary that meet individual demands of the embedded systems world. Therefore, we developed and evaluated the SPES modeling framework as a potential solution to the challenges arising in embedded system development. The project was a success, as challenges such as complexity management, which was previously thought of as an insurmountable obstacle, is now one of the core SPES principles and has become an indispensable tool for profitable engineering of embedded systems.

*Lessons learned in developing the SPES modeling framework*

The SPES modeling framework is a milestone for the model-based development of embedded software in the industry. The most striking factors that contributed to the success of the SPES modeling framework can be summarized as the following lessons learned:

- ❑ The multitude of systems gives rise to a multitude of engineering challenges.
- ❑ Model-based software development is increasingly important.
- ❑ Integrated development is essential for the engineering of embedded systems.
- ❑ Interdisciplinary knowledge networks foster innovation.
- ❑ We have achieved a lot — but a lot more still remains to be achieved.

In the following sections, we will give a short discussion of each lesson learned.

## 17.1 The multitude of systems gives rise to a multitude of engineering challenges

*Not all embedded systems are created equal*

The term *embedded system* appears to describe a relatively clear-cut type of system. However, on closer inspection, it becomes obvious that embedded systems are more likely to be a class of systems that comprise technical as well as nontechnical aspects, such as business model, product properties, problem classes they address, context conditions they may encounter, etc. In other words, the term *embedded system* may describe a system that has multiple systems embedded within itself, such as a cockpit in the avionics domain or a rolling mill in the automation domain. On the other hand, the term *embedded system* may also refer to a system that is embedded within an environment of other systems, such as an engine control unit in the automotive domain.

In addition, dominant product properties may vary in aspects such as criticality, safety, variability, real-time constraints, etc. The problem classes embedded systems are meant to address may include continuous closed-loop control tasks, data-centric computing problems, or user-centered interactions.

These aspects lead to different processes in the engineering of such systems and result in vastly different architectures and solution approaches. However, there are also some commonalities that may be regarded as conceptual building blocks for specialized development concepts. Some of these building blocks have been translated into SPES principles and were addressed in Part II and Part III of this book.

## 17.2 Model-based software development is increasingly important

*Increasing volume of embedded systems drives need for cost-efficiency*

Embedded systems development is a lucrative business with total revenues ranging to billions of euros. The number of embedded systems in the world is steadily increasing as these systems have become an integral part of our daily lives. For example, vehicle systems that were previously entirely mechanical, such as the braking system, are more and more often being implemented by electronic means using embedded systems [Volpato 2004]. Furthermore, due to the increasing interoperability of functions and features, system complexity is steadily increasing.

In order to remain profitable, development of such systems must be cost-efficient, deal with increasing system complexity and increasing quality demands, and do so in short time to market. However, when faced with the challenges of modern embedded system development, traditional development methods are lacking with regard to these aspects. In particular, meeting high quality demands is impaired by the increased product complexity.

Therefore, novel development paradigms must be established in the industry. One such paradigm is model-based software development: it promises increased productivity and therefore faster time to market, increased quality due to constructive consistency, and improved manageability of system complexity through abstraction. It is therefore reasonable to conclude that model-based software development is a core technology for the development of embedded systems.

*Model-driven development in industry and academia*

As part of the SPES 2020 project, a survey was conducted (see Section 16.3) to characterize the significance of model-driven development (MDD). The results indicate that model-based development is not only economically relevant, but is also expected to introduce major benefits for both industry and academia. In the SPES 2020 project, we have laid the foundation for successful model-based engineering of embedded systems; however, there is still much to be done.

## 17.3 Integrated development is essential for the engineering of embedded systems

Model-based development promises many benefits for the engineering of embedded systems. However, continuous development by means of integration of various engineering activities still remains essential. This means that artifacts must be continuously elicited, documented, modified and refined, starting during requirements engineering, via the modeling of system functions to their final deployment on the embedded system, across many layers of abstraction from system to subsystem to component. In addition, these development artifacts must be shared with other facets of development, such as safety and security engineering, mechanical engineering, etc.

*Development continuity is necessary*

The SPES modeling framework provides a basis for such development continuity, as described in Chapter 3. Initial evaluations show promising results (see Chapters 11 through 16), but also show that tailoring of the SPES modeling framework is necessary in almost every development context. Therefore, more evaluations and additional case studies must be conducted in order to give a better picture of when tailoring is necessary and how the engineering process can be guided such that the utmost benefit of model-based engineering can be gained.

## 17.4 Interdisciplinary knowledge networks foster innovation

*Different domains produce different types of systems and requirements*

As shown in Section 17.1, a multitude of different types of embedded systems causes new and intriguing challenges for development and fascinating avenues for research. These challenges and research avenues cannot be tackled in solitude. Instead, interdisciplinary networks of partners that ordinarily compete on the international market within and across application domains allow sharing of different perspectives and insights into the state of practice in engineering of embedded systems. Working in close cooperation with research institutes across Germany and the world fosters innovation in research, quick knowledge transfer from research to industry, and enables evaluation of research results in industrial settings.

## 17.5 We have achieved a lot — but a lot more still remains to be achieved

*Future work*

In spite of the significant innovations of the SPES 2020 project, there are a number of areas that remain to be addressed. Some of these can be summarized as follows:

- *Engineering process prerequisites:* We have seen that tailoring of the SPES modeling framework is necessary in most development contexts. As tailoring was mostly driven by the individual properties of the respective application domain, the SPES modeling framework has been tailored in quite different ways (cf. the tailoring in the automotive domain vs. the tailoring in the energy domain). Therefore, a systematic investigation of different development contexts and engineering processes must be conducted to gain insight into what prerequisites must exist in a development scenario in order for the SPES modeling framework to be applicable.
- *Variability, deployment, reuse, and other qualities:* While functional aspects and real-time and safety concerns have been considered in the SPES modeling framework, many more system qualities remain. For example, variability has not been investigated during the development of the SPES modeling framework, yet plays a major role in the engineering of embedded systems (as can be seen, in part, in Chapter 12.3.3). Therefore, additional research should focus on extending the SPES modeling framework so that these quality aspects are considered as well.
- *Engineering artifact quality:* While the SPES modeling framework allows for the development of engineering artifacts that can be used in coordination with different development activities (such as behavior models in the requirements viewpoint and the functional viewpoint), some issues remain with regard to quality assurance. How can engineering artifacts be validated as early in the development process as possible? In particular, validation must also be done with regard to quality aspects, such as real-time, safety, variability, or deployment.
- *Modular safety assurance:* As explained throughout this book, many application domains are governed by strict safety and security guidelines and standards. Although the SPES modeling framework

aims at bridging the gap between safety and security engineering and other engineering activities, one open question is how to constructively ensure, during engineering, that important standards and guidelines are fulfilled. Also, since many development projects extend existing systems by introducing some new feature or altering the existing system, further research must be dedicated to reducing the re-certification overhead, e.g., by certifying parts of the system and only having to re-certify those parts that have been modified.

## 17.6 Summary

The innovation alliance SPES 2020 has laid a solid foundation for the engineering of software-intensive embedded systems. An SPES modeling framework has been developed that is based on five principles that meet the requirements of the application domains as established in Chapter 2. The SPES modeling framework has been evaluated in the different application domains. Results show that while the SPES modeling framework meets the basic needs of industrial development, tailoring is necessary in most development contexts. Also, there are a number of research areas that have not been addressed by the SPES 2020 project, yet are important for the engineering of embedded systems: How can qualities such as variability, modular safety assurance, or optimal deployment be considered during system development? How can engineering artifacts be validated as early as possible in the development process? What prerequisites must a development process fulfill in order to be able to apply the SPES modeling framework with minimal tailoring? These are all questions that future work must address.

During the SPES 2020 project, we learned that these research questions must be answered in a collaborative way by making use of multiple perspectives from both academia and industry representatives from various domains. Only thus can research take account of the wide variety of different types of embedded systems, their roles, responsibilities, and the application contexts for which they are designed. Future work must also be spent on continuing the development of continuous, model-based engineering approaches, as these are promising approaches to surmounting the obstacles posed by steadily increasing cost pressure, increasing complexity, and the demand for high product quality in the engineering of embedded systems.

## 17.7 References

[Sikora et al. 2012] E. Sikora, B. Tenbergen, K. Pohl. Industry needs and research directions in requirements engineering for embedded systems. In: Requirements Engineering Journal, Vol. 17, No.1, 2012, pp. 57-78.

[Volpato 2004] G. Volpato: The OEM-FTS relationship in automotive industry. In: International Journal of Automotive Technology and Management, Vol. 4 No. 2-3/2004, 2004, pp- 166-197.