Ottmar Bender
Martin Hiller
Bastian Tenbergen
Dr. Thorsten Weyer

# 2

# Requirements from the Application Domains

*This section provides an overview of the current situation of embedded systems development in the individual domains that participated in the SPES 2020 project. Then, based on the current situation, the section develops concrete requirements for a continuously model-based development process for embedded systems for each domain. The section concludes with a report on an empirical analysis of these requirements.*

K. Pohl et al. (eds.), *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*,
DOI 10.1007/978-3-642-34614-9_2, © Springer-Verlag Berlin Heidelberg 2012

## 2.1 Initial Situation in the Application Domains

*Embedded systems are different in each application domain*

Embedded systems are different in every application domain. These differences arise due to the contexts in which the embedded systems will be deployed. For example, while an engine control unit is embedded in a car, a pacemaker is embedded within a human being. However, not only the context of the embedded systems, but also the constraints under which development takes place are different in each domain, even for each development project. For example, in the avionics and healthcare domains, very strict safety standards and certifying authorities govern the process as well as the product to be developed.

On the following pages we discuss initial situations within each application domain. The intention is to shed some light on current development situations in which embedded systems are being developed. Based on these situations, concrete requirements for a continuous model-based development process can be elicited, as discussed in Section 2.2.

### 2.1.1 Initial Situation in the Automation Domain

In the automation domain, embedded systems can be found at different levels. There are *devices for automation,* including intelligent sensors and actuators, that are typically presented and offered to the customer in a product catalog. On the one hand, there are standard devices such as programmable control devices or standardized power trains. On the other hand, there are also numerous special purpose devices for specific problems in automation: *machines (e.g*., *robots)* that are either provided to the user in a product catalog or developed individually for customers, or *partial constructions* (e.g., a product line) or entire constructions that are developed individually for a specific user.

*Bespoke system development*

Hence, embedded systems in the automation domain are both: systems that are primarily realized in development projects independent of any customer contract and systems that are realized primarily in projects based on a contract with the customer. In projects with customers in particular, the usage view of how an entire embedded system is developed by means of embedded software-intensive systems and which concepts the automation devices provide to support the integration into an entire system is of vital importance.

For a *provider of automation devices*, this results in the following challenges with regard to business and technology:

- Managing the numerous variants that result from different "performance parameters," bus systems, requirements for safety, reliability, etc.

- Ensuring high quality and robustness
- Ensuring that the automation devices are capable of being integrated with each other as well as with other devices

For a *system integrator,* this results in the following challenges with regard to business and technology:

- The entire construction and essential parts of it are developed within projects with customers, which, based on experience, entails a high corporate risk.
- The development process comprises the integration of several purchased components and services, which entails a high technical risk.
- During development, different disciplines such as process technology, mechanics, electrical engineering, and software must be integrated with respect to the procedure and the work results. Since the software is integrated in the last step, this integration has the task of ensuring the correct interplay between the disciplines.

### 2.1.2 Initial Situation in the Automotive Domain

*Steadily growing complexity*

Embedded automotive systems built for today's mobility requirements are growing in complexity. Therefore, new and refined development methods are required. Across all domains, engineering for embedded systems is characterized by a physical context with real-time requirements and the need for interdisciplinary cooperation. Additionally, the automotive domain has a high proportion of quality requirements, cost pressure, and resource constraints. These result from high volumes ranging in the millions, particularly demanding safety and reliability requirements, and extensive variability stemming from a large number of system approaches and functional configurations.

Within the SPES 2020 project, the partners in the automotive domain planned to address the challenges described above with the objective of developing new or refining existing methods that ensure that systems with this level of complexity can be developed more efficiently.

### 2.1.3 Initial Situation in the Avionics Domain

Embedded avionics systems built for today's aircraft are growing in size and complexity, therefore new and refined development methods are required. To improve this situation, the SPES 2020 project was established. SPES 2020 aimed at developing new or refining existing

methods to ensure that systems growing in size and complexity can be developed more efficiently.

*Strict certification guidelines and safety requirements*

Avionics systems have to be certified by the airworthiness authorities before they can be installed and operated in an aircraft. In order to achieve certification for an avionics system, the company developing and manufacturing this system has to provide the airworthiness authorities with the evidence that the system is safe.

A system is considered safe if two prerequisites are fulfilled by the system builder: firstly, a safety analysis has been conducted that shows that the probability of hazards caused by the system is sufficiently low; secondly, the manufacturer can demonstrate that the system's hardware and software parts have been developed according to pre-defined processes and have been accepted by the airworthiness authorities. This means that methods developed or refined in SPES 2020 had to support and comply with the safety analysis and development processes in the avionics domain. The methods developed in SPES are applied in the typical development process steps: *requirements analysis*, *design*, *implementation*, *integration*, *verification,* and *validation*.

Current methods and principles of incremental certification also have to be refined to reduce certification effort for future systems. On the other hand, efficient methods are necessary for recertification of systems that have been certified once and have to be modified, e.g., due to implementation of new features.

### 2.1.4 Initial Situation in the Energy Domain

*Networks of heterogeneous systems*

It is widely understood that modern power generation will consist of a mix of generation from conventional power plants, such as nuclear or coal-fired power plants, and an increasing amount of generation from renewable energies such as wind, biomass, and solar power. The latter part of the generation mix is installed in a decentralized manner, i.e., apart from some larger power generators such as wind turbines, there are plenty of small generators with approximately 5-100 kW. Examples are photovoltaic units placed on the rooftops of residential homes or combined heat power generators.

Besides the trend towards a large number of small energy resources, we can observe that more and more often, the characteristics of devices such as inverters towards the distribution grid can be controlled electronically and are programmed into the devices themselves. This raises the question of how to set up and evaluate a massively distributed energy system taking advantage of the benefit of the “magic of large numbers” in the case of small, controllable energy resources. Due to the

small size of the distributed energy resources, the characteristics that have to be considered for managing such distributed energy systems are:

- More than 100 resources have to be managed simultaneously.
- The energy resources are often operated by local operators with local objectives (dual use).
- Engineering effort per resource is not affordable.
- Available communication infrastructure including power line communication (PLC) should be reused.
- There is no guarantee that a resource is operating in a controlled fashion.

Based on the characteristics, the following technical requirements have to be fulfilled by an implementation managing the system:

- Full self-configuration of the distributed energy resources
- Tracking of availability and operation of energy resources
- Generic modeling of the different energy resources
- Compensation of stochastic loss of control of particular energy resources
- Requirements of the particular states of the distribution grid

### 2.1.5 Initial Situation in the Healthcare Domain

Similarly to the avionics domain, embedded systems in the healthcare domain are governed by strict safety guidelines and standards and are required to pass certification before they can be legally operated. In particular, rules imposed by regulatory authorities such as the FDA not only have to be adhered to by the product, but in some cases by the development process as well. As a consequence, safety and regulatory concerns dominate the development process and must be considered meticulously during system development.

*Safety aspects within medical systems residing in the human body*

Furthermore, some types of embedded systems in the healthcare domain are particularly constrained, as they reside within a human body. Pacemakers, for instance, are subject to special constraints regarding their maintainability. Once the system is installed, changes, maintenance, or alterations are undesirable, as they would require the patient to undergo further, potentially dangerous surgery. Therefore, it must be possible to maintain these systems easily and ideally externally (if at all). In summary, systems that reside within the human body have properties that must be accounted for during development to ensure safety, maintainability, and testability.

## 2.2 Requirements for the SPES Engineering Approach

The SPES engineering approach is intended to address the challenges that arise in the application domains. These challenges were outlined in Section 2.1. The requirements for the SPES engineering approach are presented below. These requirements are based on the initial situations presented above.

### 2.2.1 Requirements from the Automation Domain

Automation must cover process technology, mechanics, and electrical engineering in the individual projects to different degrees. Therefore, the requirements for the software are very different (see, for example, the diverse security standards). From the perspective of the automation domain, the following requirements and challenges were therefore addressed within the SPES project:

- *Supporting the modeling of technical systems:* Due to the increasing complexity of systems in the automation domain, it is necessary to create adequate models in order to be able to manage the development of these systems. For this purpose, the following means are commonly used: appropriate abstractions, different views on the system, consideration of different aspects. In addition to the individual models themselves, dependencies between the different models have to be documented as well as the development process, or more specifically, the underlying development methodology used to create or refine the models during development. In order to be able to exploit all the benefits of model creation, an appropriate theoretical foundation of the modeling approach is necessary.
- *Supporting system integration:* Due to the necessity of integrating the different systems and services when developing systems in the automation domain, the specific aim is to use the models to support this integration. This involves not only compatibility of the content of the models, but also the procedure with regard to the engineering workflow during development of the systems. Considering the systems thereby merely from a software-technical view is not sufficient, since the software is embedded in a physical system and has to be considered as an integral part of it, because typically the architecture of the software is determined by the architecture of the physical system.
- *Ensuring specific system properties right from the beginning:* To reduce the technical and business risks faced during development of

systems in the automation domain, the models shall be used to ensure essential system characteristics. Therefore, an appropriate model of the physical system in which the software is embedded is necessary (as well as the model of the software). In this way, the system can, for example, be put into operation virtually. This also requires an appropriate open infrastructure for executing the models.

- *Providing tool chains for engineering of embedded systems in the automation domain:* Due to the complexity of systems in the automation domain, their development must be supported by adequate engineering tools. Since a number of different tools are already used in practice, a main requirement was to extend these existing tools appropriately, to use them, and to integrate them into a continuous tool chain. The integration shall be driven by a consistent modeling theory for technical systems to ensure sustainable tool support.

### 2.2.2 Requirements from the Automotive Domain

The major requirements from the automotive domain for the SPES engineering approach are the following:

- *Supporting the systematic gathering and documentation of requirements:* The SPES modeling framework shall apply model-based requirements engineering to the automotive domain systems to enhance system understanding, provide guidance for system design, and deliver proof of fulfillment of required properties.
- *Supporting the transition between informal and formal requirements:* The SPES modeling framework should provide a systematic method for transforming a set of mainly informal requirements into an implementation that is based on the domain-specific AUTOSAR standard.
- *Supporting functional development:* The SPES modeling framework shall introduce model-based functional development throughout the automotive development lifecycle and shall integrate the discrete and continuous problem classes into a homogenous system design.
- *Supporting model-based safety design:* The approach should support model-based safety design in automotive development to achieve safety properties by design and reduce the safety validation effort.
- *Supporting the analysis of functional correctness:* The SPES modeling framework shall identify design flaws regarding functional correctness and timing in early development phases.
- *Supporting the AUTOSAR standard:* The SPES modeling framework shall transform the SPES metamodel into corresponding AUTOSAR

models in order to apply the SPES analysis techniques in the AUTOSAR context.

- *Empirical validation:* The SPES modeling framework shall validate and explore the limits of the developed methods in the automotive domain empirically to prove the effectiveness of the approaches.

### 2.2.3 Requirements from the Avionics Domain

The major requirements from the avionics domain for the SPES engineering approach are the following:

- *Supporting the systematic specification of requirements:* To specify consistent, understandable, and unambiguous system and software requirements, the application domains in SPES 2020 proposed to elaborate available formal or semiformal specification languages to ensure their efficient usability. An important aspect of usability is that requirements must be readable and comprehensible to engineers and representatives of certification authorities. To improve the understanding of the dependencies among the requirements, the need for a suitable modeling technique was identified for SPES 2020. A further objective in this area was to exploit the formalism in the requirements in order to generate test cases or test case fragments automatically and to utilize the formal character of the requirements to perform consistency and completeness checks on the requirements specification. The requirements engineering methods developed must adhere to different levels of certification strictness imposed by the different certification standards and authorities.
- *Supporting the systematic analysis and documentation of system architecture:* In order to develop large scale systems, for example, smart grids, rolling mills, and aircraft, a systematic refinement of architectural modeling techniques that takes different aspects (e.g., safety) into account is required. In addition, the definition of abstraction layers and the relation of artifacts between them is very important for coping with the complexity imposed by the scale of the systems considered. The optimization of possible design solutions regarding safety and performance and the utilization of the computing resources available has been stated as a further goal in SPES 2020. It has been recognized that the availability of a modeling language (e.g., SysML) is not sufficient to achieve the goals described above. In addition, efficient modeling techniques and methods are required.
- *Supporting continuous modeling of safety and system certification:* Safety and system certification play an important role in the

application domains of SPES 2020, especially in the automotive, avionics, and healthcare domains. Today's safety analysis is performed on separate design and safety models using different tools. To avoid error-prone and inefficient redundancy of the design and safety models, a requirement for the SPES 2020 program was to define solutions allowing a safety analysis that integrates system design and safety information in one model. A further requirement for the avionics domain was to elaborate methods for automatically generating safety cases based on the system design model.

- *Supporting the verification of engineering artifacts:* The verification activities create a high workload in the application domains. The requirement for the SPES 2020 program was to reduce this workload by defining methods for generating test cases and procedures automatically based on requirements and design information.

### 2.2.4 Requirements from the Energy Domain

The major requirements from the energy domainfor the SPES engineering approach are the following:

- *Supporting the consideration of large numbers of massively distributed embedded components:* Smart grids are potentially large systems with huge numbers of massively distributed embedded components (up to the order of millions distributed across hundreds of square kilometers). The high number of components in real systems can cause the overall system to show characteristics that do not emerge in smaller systems with a comparatively low number of components.
- *Dealing with complexity in system structure and component interaction:* Certain events in a power grid, for example, a decrease in generated power, can cause events within the communication network, such as messages, to switch off consumers. Power generation can be affected by weather conditions (solar power, wind power). Consumer behavior influences energy demand. Energy markets and international integration of power grids influence energy transmission and financial transactions.
- *Supporting the engineering constraint "one-shot scenario":* Because of the sheer size of smart grids, many technical decisions that are taken during the concept phase are virtually irreversible after the smart grid has been realized and installed. Thus, careful planning before starting the realization phase is very important. Mistakes can lead to huge costs in later project phases. This is why it is important

to design and test smart grids by means of simulation before installation to uncover and fix problems beforehand.

- *Supporting dynamics in system structure and system behavior:* Due to the characteristics and the implementation of specific components within a smart grid, system structure and system behavior can be dynamic, for example, with respect to availability, stability, and failure resilience. For example, components within a smart grid, such as local generators and consumers, can join or leave the grid at any time. Furthermore, due to the massive distribution of components, no single entity has total control of all components in the smart grid. Failure or faulty behavior of single components within the system is thus inevitable. The overall system must be able to cope with these challenges.

### 2.2.5 Requirements from the Healthcare Domain

The healthcare domain faces similar challenges to the avionics domain. Most embedded devices in medical applications are safety-critical and have to pass a long and intensive certification procedure. Again, similar to avionics, in such systems a safe state cannot be reached by simply switching off the equipment. Imagine, for example, a life-supporting device where the health of the patient depends critically on the correctness of the embedded software. Thus, all components must be redundant and highly reliable. The software development process for such a system has to follow strict rules and all artifacts must be validated.

For a model-based software development, this means:

- *Safety aspect:* Safety is of utmost importance. It must be possible to explicitly state safety requirements in the models and to assess whether they are realized in the final system. Other nonfunctional properties such as usability, adaptability, and configurability can be regarded as special instances of safety.
- *Traceability:* In order to allow for effective validation, all models must be linked to each other such that requirements can be traced from the initial specification, through the various modeling stages, to the final executable code. Ideally, the models are instances of a common metamodel and have clear, unambiguous semantics.
- *Interoperability and adaptability:* Devices and processes must be interoperable and easily adaptable. This can be achieved through standardized reference architectures. The modeling methodology must provide a way of including such reference architectures (e.g., as a model library) and instantiating the standardized component for a specific project.

- *Energy efficiency and maintainability:* Since devices such as pacemakers remain within a patient for a long time, energy efficiency and low maintenance efforts are very important for certain medical devices. The methodology must provide ways of combining resource considerations with the certification needs as required by regulatory descriptions.
- *Testability:* Testability is an important assessment criterion. Therefore, the modeling framework should support automated test case derivation from models as well as code generation. There must be a clear distinction between implementation models and test models.

Model-based design has a high potential to improve the software development process of embedded medical devices. However, software development is only one aspect in the design and production of such systems. The SPES modeling framework shows how validation and certification aspects can be incorporated into the process to allow for better products and a significantly shorter time-to-market.

## 2.3 General Requirements from Industry

At the beginning of the SPES project, a study was conducted with representatives from companies in all application domains of the SPES project in order to gather their major requirements from industry concerning the SPES engineering approach. The participants' self-reported areas of operation included research and development (40% of the participants) as well as process and project consulting (another 40%). Of the participants questioned, 60% reported their experience with requirements engineering to span 5 to 10 years, 20% even reported more than 15 years of experience, and 90% of the participants reported their level of experience in requirements engineering as advanced or expert.

The study employed a combination of qualitative and quantitative techniques in order to yield deep insight into the state of practice and the needs concerning model-based engineering. Data was acquired by means of a structured interview and a post-interview questionnaire.

The findings from the study are summarized below and related to the focus of the SPES 2020 project. Detailed information about the motivation for the study, the feedback gained from the participants, and the detailed analysis and conclusions can be found in [Sikora et al. 2012].

### 2.3.1 Empirical Finding: Need for Model-Based Engineering

*Natural language requirements vs. requirements models*

Natural language is the most common documentation form for engineering artifacts. However, there is strong evidence that practitioners are dissatisfied with natural language, as dealing with large bodies of natural language documents is perceived as tedious and error-prone. In contrast, using models during the engineering of embedded systems is perceived as beneficial as models help in the understanding of complex engineering problems, serve as a natural means for structuring the problem space, and make communication with other stakeholders easier. As a result, in current practice, models are used to support engineering and often supplement text-based documentation of engineering artifacts. Our study showed, for example, that executable models (such as MATLAB/Simulink models), semiformal models (such as SysML [OMG 2010a] and UML [OMG 2010b] models), as well as domain-specific models from disciplines such as mechanical engineering, electrical engineering, or control engineering are common artifacts throughout the engineering process.

*Artifact-based quality assurance*

One of the most important purposes of these artifacts is early validation and quality assurance. However, despite the advantages of using models, many practitioners refrain from applying them. One key reason is that there is confusion about when to apply models during engineering and when to resort to traditional natural-language-based documentation, particularly when legally binding documents are involved, safety standards must be satisfied by means of models, or models are used that are applied in different engineering activities (for example, structural models that are used during requirements engineering as well as architecture design).

In summary, an engineering approach is needed that fosters model use during different engineering activities and supports the use of model types that are already common in the engineering of embedded systems.

### 2.3.2 Empirical Finding: Need for Artifact-Orientation

*Artifact interoperability*

As mentioned in Section 2.3.1, the state-of-practice study was able to confirm that natural language is the dominant documentation form for engineering artifacts. As a result, artifacts in the engineering of embedded systems are typically natural language documents that also serve as a contractual basis and are at best supplemented with models. Furthermore, as these documents are usually holistic in nature, information contained therein for specialized engineering activities such as quality assurance, architecture design, or safety engineering is hard to

discern. In particular, many engineering artifacts in current practice do not meet the prerequisites for applying automated techniques.

Therefore, approaches are needed that allow for the co-development of artifacts that can be used and re-used for a variety of engineering activities.

### 2.3.3 Empirical Finding: Need for Continuous Method Support

As mentioned in Section 2.3.1, there is uncertainty about when to use models to aid the engineering of embedded systems, and this is one of the key factors inhibiting more intensive model support. Another inhibiting factor is missing method support for the application of models during different engineering activities.

*Artifact integration between engineering activities*

While some knowledge exists regarding different model types and their suitability for different engineering activities, an approach for the seamless integration of models during engineering and the transition between engineering activities is largely missing. For example, missing method support leads to an enormous effort for ensuring the consistency between requirements engineering artifacts and safety engineering artifacts. In particular, method support is missing for specifying artifacts across a hierarchy of abstraction layers. In addition, results of the study provide evidence for a close interrelation of requirements engineering and architecture design, but also indicate some confusion regarding the separation of the resulting artifacts from both engineering activities (see Section 2.3.1). As a consequence, participants expressed a strong need for systematic support for traceability between these two engineering activities.

### 2.3.4 Empirical Finding: Need for Differentiation of Abstraction Layers and Transition between Them

Since the complexity of modern embedded systems is continuously increasing, new challenges for their engineering also arise. In order to meet these challenges, the development process must be structured strictly. The participants of the study stated that performing engineering across a hierarchy of abstraction layers is one of the essential means to achieving a structured development process. In particular, a systematic approach that takes the refinement of engineering artifacts into consideration is missing from current practice.

*Seamless transition and integration*

In addition, practitioners expressed the need for seamless transition between abstraction layers. Although abstraction layers are seen as

beneficial, confusion exists concerning their application. In particular, there is uncertainty about which engineering artifacts to define at what level of abstraction, what level of detail should be included in an artifact, how abstraction layers can be tailored to specific project needs, and how consistency between artifacts of different abstraction layers can be maintained.

*Tailorable abstraction layer hierarchy*

As the results of the study illustrate, the application of abstraction layers is not standardized in industry, is highly influenced by the application domain (e.g., automation, avionics, or healthcare), and varies depending on the engineering context (e.g., the specific system type, properties of the supplier-integrator relationship). In some cases, the use of abstraction layers is formally imposed by standards.

In summary, the study showed that the application of abstraction layers in industry depends largely on the engineering context and in particular on the responsible engineers' intuition and experience. Therefore, improved method guidance for specifying artifacts across different abstraction layers of an embedded system were needed at the beginning of the SPES project.

## 2.4 References

[OMG 2010a] Object Management Group: OMG Systems Modeling Language™ (OMG SysML) Language Specification v1.2. OMG Document Number: formal/2010-06-02.

[OMG 2010b] Object Management Group: OMG Unified Modeling Language™ (OMG UML), Infrastructure v2.3. OMG Document Number: formal/2010-05-03.

[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.