

Challenges in Engineering for Software-Intensive Embedded Systems

As long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now that we have gigantic computers, programming has become a gigantic problem.

Edsger W. Dijkstra, ACM Turing Award Lecture, 1972 [Dijkstra 1972]

1.1 Core Value of Embedded Systems Development

Embedded systems in everyday life

It is hard to find another market in information technology that shows similar steady growth to the market for embedded systems. As microcontrollers, they have taken over a variety of functions in multitudinous technical systems, for example, in manufacturing plants, medical equipment, power supply systems, aircraft, and cars, but also in home appliances such as washing machines and refrigerators. Microcontrollers monitor and control the systems they are embedded in. In doing so, they interact directly with their environment via communication devices or indirectly via sensors that capture data such as temperature or movement, as well as with actors that transform those data into action. Embedded systems play a key role in many high-tech sectors: they are essential in modern transportation systems — from cars, to railway vehicles, to aircraft. They automate manufacturing plants for all businesses, and are an integral part of powerful medical equipment.

Market for embedded systems

Embedded systems are microcontrollers that are connected to complete systems via sensors, actors, operator controls, and communication devices. They interact in various ways with their environment, offering a variety of functions through comprehensive software. Ninety-eight percent of the microcontrollers produced worldwide are employed in embedded systems. The programs that are executed by these embedded systems, known as “embedded software,” represent an essential part of these systems and define their functionality decisively.

Embedded systems as innovation drivers

The “Nationale Roadmap Embedded Systems,” issued by the German “Zentralverband Elektrotechnik- und Elektroindustrie e.V.” (ZVEI) in 2008, forecasts an average annual growth of 9-10% for the embedded systems market in the coming years [ZVEI 2008]. The estimated volume of the worldwide market for embedded systems amounts to around 60 billion euros. For the most important part of embedded systems, the software, the study shows a significantly higher growth than for the hardware part.

Key competence

In businesses highly affected by mechanical and electrical engineering, software has become the most important innovation driver: a current mid-range car employs more than 70 embedded systems. The functions of the anti-lock braking system or the controls of the engine’s ignition point are determined by the software of the respective embedded systems.

The capability of developing high-quality, target-driven embedded systems is a competency Germany needs in order to maintain and develop a leading position in economically important industry sectors such as vehicle manufacturing/automotive engineering, aviation, plant engineering, and automation technology, as well as medical engineering. Germany's export strength is mainly on technologies in which embedded systems represent a core value. The control of increasingly powerful and extensively networked, and as a consequence more complex embedded systems is a huge scientific and technical challenge. Mastering these crosscutting competencies offers opportunities in many application areas. Shortcomings, however, inevitably lead to risks and finally to loss of markets. It is essential, therefore, that Germany positions itself strategically in this field.

Germany is well known for the high quality of its products, mainly in areas such as automotive engineering, aviation, automation technology, and medical engineering, where embedded systems are deployed. To maintain this image with respect to embedded software as well, we have to make the same high demands on embedded systems as on other technical systems with the seal of quality "Made in Germany."

1.2 The Future of Embedded Systems

The future of embedded systems is determined by several trends that we can already recognize today:

- ❑ Increasing computing power of the systems together with the ability to store an almost unlimited amount of data at extremely low costs. This development, which we already know from computer systems, is now finding its way into embedded systems. *Increasing computing powers*
- ❑ Future embedded systems will be more and more networked. Networking mainly via the Internet, arguably the most important development since the invention of the letterpress, but also using a wide variety of different networking technologies, will multiply the intrinsic intelligence of embedded systems *Networked systems*
- ❑ The increasing structural and functional integration with mechanical and electrical system parts will finally produce "cybertronic systems," consisting of mechanical, hydraulic, or pneumatic parts, as well as sensors, actors, and information processing units linked to each other via flows of energy, material, or information. The line between mechanical components and software system will become more and more blurred. The software part of "embedded systems" is *Structural and functional integration*

shifting from an enabling technology towards a core technology: it forms the products to which it belongs. The software part transforms, enriches, and becomes the dominant part of the new generation of products [Beetz 2010].

High complexity Future embedded systems will be characterized by high complexity. Compared to pure mechanical or electrical systems, future embedded systems will have a much higher number of coupled elements (networking). In addition, these elements will be implemented by different technical disciplines involving different types of coupling (mechanical, electrical, IT). More and more software will be crucial for the functionality and the coupling of the individual mechanical and electrical components. This coupling will finally lead to the tight integration of today’s isolated engineering disciplines of mechanical engineering, electrical engineering, and computer science.

The development leads from closed, well-arranged embedded systems that can be found in coffee machines, ATMs, or heart pacemakers, to systems that become more and more intelligent, to systems that can build their own intentionality, enabling them to make goal-oriented decisions on their own and to act accordingly. Ancestors of those future embedded systems are already among us today as helpful servants that can cut trees for us, prepare sandwiches, assemble cars, fly airplanes, or explore distant planets.

Integration with classical information and communication technology As a consequence of the integration of embedded systems with classical information and communication technology and the Internet, systems will arise that span the globe, working together seamlessly and forming separate “digital spheres” such as global intelligent shells consisting of android buildings, factories, hospitals, transportation systems, up to highly automated agriculture and an inexhaustible knowledge base. The dream of the famous encyclopedists of the age of enlightenment *Denis Diderot* and *Jean-Baptiste le Rond d’Alembert*, to make all the knowledge of the world accessible and useable to everybody, seems to be coming true, because the Internet age has just begun.

Our knowledge is being transferred to the Internet at full speed. Eric Schmidt, former Google CEO and now chairman of the board, illustrated the tremendous storage capacity of the Internet as follows: “In 2029 you will be able to buy eleven Peta-Bytes (quite a big number) of digital storage on a single hard drive for less than 100 Dollar. According to my calculation this device will be able to store every single day, 24 hours in DVD-Video quality for six hundred years” [Schirmmacher 2011].

Digital spheres No knowledge, no experience that cannot—and will not—be recorded. However the true value of information is not based on the

information itself, but on the networking and the algorithmic evaluation of the information stored. Therefore, the “digital spheres” not only contain information and knowledge, but moreover awareness of how to use the information and the knowledge, which in turn will change the information and the knowledge permanently.

Thus, a separate digital parallel world is arising, to which the term “artificial intelligence” is probably more applicable than to the traditional field of science that is aimed at emulating the human brain. Humans will be part of this intelligent parallel world and they will have to shape and control it. We will not discuss the social and political consequences here, but they will drastically change our lives, especially our professional life and the way we live together.

Examples of these “digital spheres” are the global communication network, the huge worldwide information marketplace of the financial world, and the Global Positioning System (GPS) that has already become an integral part of transportation systems. In the near future, the power grid will also build such a “digital sphere.” This “smart grid” will be based on the most advanced embedded systems available. By virtue of distributed intelligence alone, the “smart grid” will be able to guarantee predictable stability and functionality of the electrical energy network. This development will not pass by many engineering disciplines, as something like “Google Engineering” will arise, where architecture and design decisions will be taken from the endless resources of the “digital sphere.”

In the US, the term “cyber-physical system” is used for those systems, forming the basis of the “digital spheres.” In a cyber-physical system, electronic systems are interlaced intelligently with network components and physical systems in a way that integrates the physical systems to give them new capabilities. The acatech (German National Academy of Science and Engineering) project “Agenda CPS” develops a general view on the political, economic, technical, and research challenges of cyber-physical systems. In this project, cyber-physical systems are understood in a broader sense, including issues around connecting embedded systems to global networks such as the Internet.

*Cyber-physical
systems*

Embedded systems are the important building blocks of cyber-physical systems and are used by the various digital spheres. Here they implement in particular the interfaces of the “digital spheres” to the user and to the technical and physical components, thereby forming the link between the virtual “digital spheres,” the humans, and the real world. Mastering the complexity this introduces constitutes a central challenge. In his keynote speech at the ITEA (Information Technology for European Advancement) symposium 2010, the Chairman of ITEA,

Rudolf Hagenmüller, coined the phrase “embedded hardware.” This comprises hardware in the primary sense, i.e., not microcontrollers or control units, but also in the sense of the scenarios described above. Embedded hardware could be a car, a building, a milking machine, or even a complete power plant — therefore, hardware that can be understood as embedded in a complex network, in a digital sphere.

The foundation for mastering the challenges of the development of embedded hardware is established by the SPES 2020 project [BMBF 2009].

1.3 Vision of SPES 2020

National innovation alliance

As outlined above, developing software for the increasingly more powerful, more internetworked, and as a consequence more complex embedded systems is a huge challenge. Therefore, 21 partners from industry and science have formed the national innovation alliance “Software Platform Embedded Systems 2020,” targeted at making the production of embedded software across industry domains professional by means of an integrated and powerful methodology [Beetz 2010, Broy 2010].

The vision of SPES 2020

Mastering the related challenges represents an important advantage for German products in European and international markets, and is therefore essential for job creation and welfare. This highlights the enormous benefit of such a concentration of research and development work, especially because of the numerous application areas in key German industries. It is the vision of SPES 2020 that in the near future, it will be possible to develop embedded systems, containing a high amount of embedded software, using a set of integrated modeling techniques whose interdependencies and cooperation are completely understood. SPES 2020 envisions that:

- ❑ The functional and nonfunctional requirements of such systems can be completely modeled at system level using appropriate abstraction.
- ❑ Analysis, verification, and validation steps can be performed based on those models.
- ❑ Decomposition for the interface behavior of the systems in the sense of architecture and a more step-by-step realization of a technical architecture can be derived from these functional models.

Uniform modeling techniques

We will still split the systems into mechanical, electrical, and technical parts. In doing so, we will use uniform modeling techniques for all three disciplines, or at least clearly defined, standardized interfaces that cover

all aspects of these three different system parts and describe their interactions in a modular fashion and in the sense of a compositional modeling technique.

The main goal is to formalize the models to the extent that a high degree of automation is possible, including consistency checks and generation of validation methods to make the systems quantifiable and their properties explicitly represented and documented. These methods can be either tests or logical analysis. The planned high degree of automation will make it possible to generate software out of its abstract and generally modeled properties for different and sufficiently specified platforms. This of course requires the modeling and specification of the platform properties in sufficient detail.

High degree of automation

The longer-term vision is the availability of a comprehensive concept for reuse for the respective application areas. Based on given, reusable platforms and system building blocks, it should be possible to execute major parts of the development work by utilizing predefined and domain-specific specifications, building blocks, and reference architectures.

Concept of reuse

1.4 Mission of SPES 2020

The national innovation alliance “Software Platform Embedded Systems 2020” (SPES 2020) thematically follows the aim of professionalization of a cross-domain development process, mainly the classical targets of software engineering: productivity and quality.

Cross-domain development

The focus of the research and development work is in embedded software, which is widely ramified into the areas of mechanics and electronics in order to leverage the comprehensive optimization potential arising from those disciplines. The goal of SPES 2020 was to create a unique innovation alliance in Germany that works across application domains to develop future networking, hardware, and software architectures, as well as new methods for software and system engineering.

A model-driven and tool-supported approach that is based on a strong mathematical foundation allows for the efficient development of embedded systems, starting with initial customer requirements, through specification of architectures, through implementation, to system verification and certification.

Integrated model-driven and tool-supported approach

This objective required a lot of work to be done in applied research as well as in fundamental research to complete relevant results. SPES 2020 was able to provide an integrated approach for model-based development of discrete systems with a strong emphasis on interfaces, distribution,

and interaction, as well on a consistently architecture-centric approach that includes adequate system structuring and a model of the individual architecture elements.

Joining the different modeling approaches

The basic terms, concepts, and theory have been available in results of fundamental research that has been more or less complete in that area. SPES managed to join the different modeling approaches supported by the academic partners. However, the main focus of SPES has been the transformation of those modeling approaches into usable terms and characterization techniques needed for an engineering approach, and the consolidation of methods for the creation of the various model views for an integrated use for target-oriented system and software development. The different, more pragmatic approaches of the application domains have been recognized.

Domain-specific approaches

The domain-specific approaches have been simplified and consolidated as far as possible, and a solid understanding has been developed with regard to which application domains really require specific modeling techniques and how advanced modeling approaches in the different domains can be brought to the general modeling approach.

Therefore, the SPES central project has taken up the available approaches from the application domains and merged them with results from scientific research. The resulting modeling methods have been reflected back to the application domains using case studies. Additional requirements from the application domains have been collected and have been considered in the definition of the modeling methodology.

Properties of the environment

A special challenge that was not the main focus of SPES was the modeling of system properties that are not classical software properties but properties of the environment (e.g., electronic or mechanical) in which the systems are embedded. In the SPES 2020 context, we restricted the work to the question of which properties of the environment are necessary, what the interfaces to the domain models look like, and how those models should be brought in to guarantee consistency between the physical models and the models of the embedded software.

A special focus in the project was on the integration of model-based development in order to leverage the synergy that comes from integrating the models across a larger number of development steps instead of looking at the models of each step in isolation.

Architecture as central artifact

Architecture is the central artifact in model-based development. In SPES, the term “architecture” is understood very broadly as any kind of structuring system from a functional view, logical component view, or technical view, with a focus on software and hardware architecture, as well as on deployment and scheduling.

Last but not least, an important goal of SPES 2020 was to provide an adequate concept for tool support based on the integrated approach for model-based development that was developed. Prototypes of this concept have been developed and tested.

With these achievements, the innovation alliance SPES 2020 has significantly strengthened the strategic and complete position of Germany as a leading country for the development of embedded systems. But to maintain this position as a leading engineering country and create new jobs and welfare for society, further effort is necessary from all stakeholders, from academia, from industry and from public authorities. Examples of future topics to be addressed by the innovation alliance are the managing of variants and multidomain engineering.

1.5 Research Approach

As discussed above, a central idea of model-based development of embedded systems is the integration of different approaches from the application domains and consolidation of these approaches to form an integrated model-based approach that has the potential to be a comprehensive tool support that can be deployed in various application areas.

By nature, this is a difficult and comprehensive task to which scientific foundations, historically grown views, and different requirements from the various application areas contribute. In addition, the participating scientific groups follow their own approaches, starting from different theories. Up until now, integration at theory level has not been completed, and a series of foundation work has yet to be done.

Against this background it was important to gain a clear vision of how the different views and approaches in SPES could be integrated. This gave four main focus points for the development of an integrated approach for model-based development.

Four main focus points

1.5.1 Generic Overall Approach — Metamodel

The generic overall approach identifies the fundamental ideas and concepts of how to proceed. These include the use of different views and viewpoints in the sense of the specification of a series of abstraction layers for modeling the architecture. These architectural layers are in documented verbally and in approximate terms, represented graphically, and finally mapped onto a metamodel. In the generic layer, the basic philosophy is described without a detailed theoretical elaboration, and

Fundamental ideas and concepts

more pragmatically, without a concrete practical implementation with respect to tool support or concrete modeling techniques. It is important that not only the philosophy of the architectural approach and its modeling framework are described, but also that the whole engineering process is captured.

Three approaches to make the generic approach more concrete are outlined below.

1.5.2 Fundamental Scientific Approach

Comprehensive theory

A rigorous, theory-based scientific approach to substantiating the generic approach is to work out a comprehensive theory of model-based development, including all theoretical investigations and elaborations, so that for all concepts and ideas described in the generic approach, there is a comprehensive scientific elaboration and theoretical foundation. This ensures that all concepts of the generic approach are completely analyzed, elaborated, and justified from a theoretical perspective.

The theoretical framework will show that all the concepts are consistent, complete, and fit together seamlessly. This lays the foundation for a methodology that is theoretical in spirit but not immediately deployable from a practical perspective, and that completely answers all questions concerning modeling and specification, not necessarily claiming that these concepts will immediately scale or can be put into practice. Consequently, this theoretical approach offers strong momentum for the concrete approach, as it forms the necessary foundation for clear terminology and the proven concepts.

1.5.3 Pragmatic Implementation

Pragmatic implementation

The pragmatic implementation of the generic approach starts with existing approaches (for example, UML) existing tools (such as MATLAB or the tools from Esterel Technologies) and tries to integrate the existing, often very fractal approaches in terms of the generic approach step-by-step. In doing so, the different views defined in the generic approach are described as completely as possible by means of specification languages and tools available in practice. A tight integration is sacrificed in favor of the use of existing tools and concepts. Where necessary, certain breaches, inconsistencies, and, to some extent, missing precision are accepted. The benefit of this approach is its possibility for a fast transfer into practice, high acceptance by practitioners, and a good adoption to the existing processes in practice. The drawback is limited automation support and limited comprehensiveness.

Of course, it is possible to utilize the scientific foundation, developed by the more theoretical SPES approaches, insofar as these can already be implemented by the existing tools. There are several advantages of this approach:

Firstly, we see faster deployment of the theory, better acceptance by practitioners, and a big impact on procedures available in practice. Secondly, SPES can benefit by discovering holes in the metamodel or different philosophies in various application areas that cannot be united easily.

The drawback of the pragmatic approach is its limited support of automation as well as a limited integration.

1.5.4 Concrete Implementation

The generic approach can be mapped to a concrete implementation without adopting the partially integrated pragmatic approaches. Starting with the generic approach, a clean theory is set up as part of the rigorous foundation and clear concepts are developed. The goal is to develop an approach that is theoretically clean on one hand, and closely aligned with the needs of practice on the other. Gaps and incompleteness are recognized and accepted and will be filled on an ongoing basis. This drives a step-by-step understanding of the approach in practice. The goal is to develop a clean practice that is always aware of its gaps and incompleteness and that can be filled slowly. In contrast to the pragmatic approach described above, a rigorous approach that only accepts and deploys clean accented methods and techniques is followed. The concrete implementation serves as an intermediate step towards a well-founded scientific implementation that puts the model framework forward step-by-step, taking the scientific foundations into account.

*Concrete
implementation*

1.5.5 Synthesis

SPES 2020 contains elements of three fundamentally different modeling approaches and relates them such that they do not compete and cripple one other, but instead have been fused in order to complementing one other. It should be noted that integration between the approaches is not possible; integration within the different approaches would be better. However, the approaches can benefit from each other by trying to translate between the different layers to highlight and use the relationship concepts.

1.6 Topics Not Addressed in SPES 2020

Advanced topics

A series of questions regarding more advanced topics related to embedded software systems have not been addressed in SPES, such as autonomy, adaptivity, and self-organization. Furthermore, questions related to future hardware architectures, such as multicore architectures, have only been explored to an extent that was necessary to verify the appropriateness of the SPES development paradigm for these new hardware architectures. Innovative architectures for embedded systems have also not been explored.

Within the research leading towards the SPES modeling framework, product line engineering was not considered explicitly. This is because it would appear that this additional dimension of variability can only be investigated once the basics of model-driven development have been comprehensively worked out.

However, the project has developed systematic methods for the architectures in the development process, such as reference architectures and architecture frameworks, as well as their role in the development process. Advanced topics, such as product line architectures and systematic reuse, are the focus of future work.

1.7 References

- [Beetz 2010] K. Beetz: Was wird besser, wenn SPES erfolgreich ist? White Paper, 2010.
- [BMBF 2009] Bundesministerium für Bildung und Forschung (BMBF), Referat Öffentlichkeitsarbeit: SPES 2020 Software Plattform Embedded Systems 2020, Infoblatt, 2009.
- [Broy 2010] M. Broy: Mission und Vision von SPES 2020. White Paper. 2010.
- [Dijkstra 1972] E. W. Dijkstra: The humble programmer. In: Communications of the ACM, Vol. 15, No. 10, 1972, pp. 859-866.
- [Schirmacher 2011] F. Schirmacher: Wir brauchen eine europäische Suchmaschine, Frankfurter Allgemeine Zeitung, 19.07.2011, No. 165, p. 27.
- [ZVEI 2008] Zentralverband Elektrotechnik- und Elektroindustrie e.V. (ZVEI): Nationale Roadmap Embedded Systems, 2008.

Model-Based Engineering of Embedded Systems

The SPES 2020 Methodology

Pohl, K.; Hönninger, H.; Achatz, R.; Broy, M. (Eds.)

2012, XII, 304 p., Hardcover

ISBN: 978-3-642-34613-2