



Klaus Pohl  
Harald Hönninger  
Reinhold Achatz  
Manfred Broy (Eds.)

# Model-Based Engineering of Embedded Systems

The SPES 2020 Methodology

 Springer

# Model-Based Engineering of Embedded Systems



Klaus Pohl • Harald Hönniger • Reinhold Achatz  
Manfred Broy  
Editors

# Model-Based Engineering of Embedded Systems

The SPES 2020 Methodology



Springer

*Editors*

Klaus Pohl  
paluno – The Ruhr Institute for Software Technology  
University of Duisburg-Essen  
Essen  
Germany

Harald Hönninger  
Robert Bosch GmbH  
Schwieberdingen  
Germany

Reinhold Achatz  
ThyssenKrupp AG  
Corporate Center Technology, Innovation & Quality  
Essen  
Germany

Manfred Broy  
Fakultät für Informatik, Lehrstuhl für Software  
& Systems Engineering  
TU München  
Garching  
Germany

ISBN 978-3-642-34613-2      ISBN 978-3-642-34614-9 (eBook)

DOI 10.1007/978-3-642-34614-9

Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012952094

ACM Computing Classification (1998): D.2, C.3, J.2, J.3, J.7

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

# Preface

Embedded systems have long become an essential part of our everyday life. They control essential features in our cars, such as airbags, braking systems, or power locks, and are used to manage our steadily increasing communication needs by means of Internet routers or cell phones. Embedded systems are essential in application areas where human control is impossible or infeasible, such as adjusting the control surfaces of aircraft or controlling a chemical reaction inside a power plant. The embedded systems industry has therefore become a multibillion euro industry.

*Embedded systems  
— opportunities and  
challenges*

The development of modern embedded systems is becoming increasingly difficult and challenging. Issues that greatly impact their development include the increase in the overall system complexity, their tighter and cross-functional integration, the increasing requirements concerning safety and real-time behavior, the need to reduce development and operation costs, as well as the need for shorter time-to-market.

Many research contributions and development methods aim to address these challenges, and theories for the seamless development of embedded systems have been proposed. However, these solutions address only a small subset of the above-mentioned problems, are only applicable in very specific settings, and lack an appropriate cross-domain validation in representative industrial settings.

*Need for an integrated  
development  
approach*

The mission of the Software Platform Embedded Systems 2020 (SPES 2020) project was thus to focus on the professionalization of a cross-domain, model-based development method for embedded systems. SPES 2020 is an innovation alliance project sponsored by the German Federal Ministry of Education and Research. In SPES 2020, 21 partners from academia and industry have joined forces in order to develop a modeling framework that is based on the latest state-of-the-art in embedded systems engineering, addresses specific development challenges, and is validated in different domains to ensure its applicability in industrial embedded systems development.

## Aim of this book

*Industry challenges,  
principles, and  
application*

The purpose of this book is to present an overview of the SPES modeling framework and to demonstrate its applicability to embedded system development in various representative industry domains. The book provides a comprehensive explanation of the basic solution concepts of the SPES modeling framework and illustrates the application of these concepts in five application domains (automation, automotive, avionics, energy, and healthcare). The book summarizes the lessons learned, outlines evaluation results, and describes how the SPES 2020 modeling framework can be tailored to meet domain-specific and project-specific needs.

## Target audience

*Researchers,  
practitioners, and  
consultants*

This book is aimed at professionals and practitioners who deal with the development of embedded systems on a daily basis. This includes developers, requirements engineers, software or hardware architects, business analysts, mechatronics experts, safety engineers, testers, and certifiers. It serves as a compendium for researchers in the field of software engineering and embedded systems, regardless of whether you are working for a research division of a company or are employed with a university or academic research institute. For teachers and consultants, it provides a sound foundation in the basic relationships and solution concepts for engineering embedded systems and illustrates how these principles and concepts can be applied in practice.

## Content of this book

This book is structured into four parts and 18 chapters:

*Status quo and  
industry requirements*

□ *Part I – Starting Point:* This part discusses the status quo of embedded system development and model-based engineering and summarizes the key requirements faced when developing embedded systems in different application domains. Chapter 1 gives detailed insight into the role of embedded systems and outlines the scope of the SPES 2020 project. Chapter 2 discusses and summarizes the requirements for the development of future embedded system development in the automation, automotive, avionics, energy, and healthcare application domains.

*The SPES modeling  
framework and its  
viewpoints*

□ *Part II – The SPES modeling framework:* This part describes the backbone of SPES 2020 — the SPES modeling framework. Chapter 3 derives the core principles of the SPES 2020 modeling framework and illustrates how these principles help in fulfilling the

requirements outlined in Chapter 2. It outlines the overall SPES modeling framework and describes the basic solution concepts of the SPES 2020 abstraction layers and viewpoints. Subsequently, Chapters 4 through 7 describe the requirements, functional, logical, and technical viewpoints of the SPES modeling framework. Each chapter defines the specific engineering artifacts subsumed by the viewpoint, outlines the basic relationships of those artifacts with the other viewpoints, and describes the engineering process across the basic abstraction layers. Chapters 8 and 9 describe how the SPES modeling framework addresses the crosscutting aspects “safety” and “real-time.”

- *Part III – Application and Evaluation of the SPES Modeling Framework:* This part describes the validation steps taken to ensure that the requirements outlined in Chapter 2 are met by the solution concepts proposed in Part II. Chapter 10 outlines the overall evaluation strategy used to assess the applicability of the SPES modeling framework. Chapters 11 through 15 describe the use of the SPES modeling framework in the five application domains automation, automotive, avionics, energy, and healthcare. Each of these chapters briefly characterizes the specifics of the application domain and shows how the SPES modeling framework can be tailored with regard to these characteristics. In addition, each chapter outlines the evaluation activities conducted in the application domain by various partners and summarizes the key evaluation results. Chapter 16 summarizes the overall evaluation results and discusses them in the context of the requirements outlined in Chapter 2 and the SPES principles described in Chapter 3. *Evaluation strategy and application details*
  
- *Part IV – Impact of the SPES Modeling Framework:* This part assesses the impact of the SPES modeling framework. Chapter 17 summarizes the key lessons learned in SPES 2020. Chapter 18 concludes this book by providing insights into open challenges for the engineering of software-intensive embedded systems. *Achieved results and future work*

For further reading, a list of relevant, advanced literature providing deeper insights is given at the end of each chapter.

## Acknowledgements

There are many people who have contributed significantly to this book.

Firstly, we would like to thank the members of the Steering Committee of the SPES 2020 project for their guidance and support

throughout the entire project and for encouraging us to document the project results in this book.

Secondly, we would like to thank Ottmar Bender, Dr. Wolfgang Böhm, Dr. Martin Feilkas, Peter Heidl, Dr. Stefan Henkler, Dr. Ulrich Löwen, Andreas Vogelsang, and Dr. Thorsten Weyer for their relentless efforts in integrating the different project activities that took place concurrently, for many fruitful discussions and suggestions, and for their critical reviews of project milestones. Much of the content of this book is a result of their devotion and attention to detail.

Thirdly, we would like to thank each and every author of the individual chapters for their patience in the book writing process, their willingness to revise their chapters time after time, and their cooperation and help in making this book a consistent and integrated product.

Last but not least, we would like to express our deepest thanks to Bastian Tenbergen and Dr. Thorsten Weyer for their tremendous effort and support in the overall editing process.

The results presented in this book have been made possible through the funding received from the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany under grant number 01IS08045. In particular, we would like to thank Prof. Dr. Wolf-Dieter Lukas, Dr. Erasmus Landvogt, and Ingo Ruhmann (all with the BMBF). In addition, we would like to thank Dr. Michael Weber of the German Aerospace Center (DLR) for supporting this project.

Furthermore, we would like to thank Tracey Duffy for her valuable language editing assistance and Ralf Gerstner from Springer for his continuous support in publishing this book.

Klaus Pohl  
Harald Hönninger  
Reinhold Achatz  
Manfred Broy

Summer 2012

# Table of Contents

<b>Part I Starting Situation</b>	<b>1</b>
1 Challenges in Engineering for Software-Intensive Embedded Systems.....	3
1.1 Core Value of Embedded Systems Development .....	4
1.2 The Future of Embedded Systems .....	5
1.3 Vision of SPES 2020 .....	8
1.4 Mission of SPES 2020.....	9
1.5 Research Approach.....	11
1.6 Topics Not Addressed in SPES 2020.....	14
1.7 References .....	14
2 Requirements from the Application Domains .....	15
2.1 Initial Situation in the Application Domains .....	16
2.2 Requirements for the SPES Engineering Approach.....	20
2.3 General Requirements from Industry.....	25
2.4 References .....	28
<b>Part II The SPES Modeling Framework</b>	<b>29</b>
3 Introduction to the SPES Modeling Framework .....	31
3.1 Motivation for the SPES Modeling Framework .....	32
3.2 Characteristics of Software-Intensive Embedded Systems .....	32
3.3 The Principles of the SPES Modeling Framework .....	34
3.4 Core Concepts of the SPES Modeling Framework.....	35
3.5 The SPES Modeling Framework .....	36
3.6 Underlying Modeling Theories.....	46
3.7 Overview of the Following Chapters .....	47
3.8 References .....	48
4 Requirements Viewpoint.....	51
4.1 Introduction to the Requirements Viewpoint.....	52
4.2 Requirements Artifacts .....	53
4.3 Integration in the SPES Modeling Framework .....	63
4.4 The Requirements Process Model across Abstraction Layers .....	66
4.5 References .....	67
4.6 Acknowledgements.....	68
5 Functional Viewpoint.....	69
5.1 Introduction .....	70

5.2	Concerns .....	71
5.3	Functional Black Box Model .....	72
5.4	Functional White Box Model .....	75
5.5	Analyses .....	77
5.6	Integration in the SPES Modeling Framework .....	78
5.7	The Functional Viewpoint Process .....	80
5.8	References .....	82
6	Logical Viewpoint .....	85
6.1	Introduction .....	86
6.2	Concerns .....	86
6.3	Logical Component Architecture .....	88
6.4	Analyses .....	89
6.5	Integration in the SPES Modeling Framework .....	90
6.6	The Logical Viewpoint Process .....	91
6.7	References .....	93
7	Technical Viewpoint .....	95
7.1	Introduction .....	96
7.2	Metamodel of the Technical Viewpoint .....	97
7.3	Mapping between Viewpoints and Abstraction Layers .....	103
7.4	How to Get from the Logical to the Technical Viewpoint .....	105
7.5	References .....	106
8	Modeling Quality Aspects: Safety .....	107
8.1	Introduction .....	108
8.2	Concerns .....	108
8.3	Component-Integrated Component Fault Trees .....	109
8.4	Efficiently Deploying Safety-Relevant Applications to Integrated Architectures .....	112
8.5	Integration in the SPES Modeling Framework .....	117
8.6	References .....	118
9	Modeling Quality Aspects: Real-Time .....	119
9.1	Introduction .....	120
9.2	Model-Based Real-Time Engineering .....	121
9.3	Modeling Platform-Independent Real-Time Requirements .....	122
9.4	Modeling Platform-Specific Real-Time Properties .....	124
9.5	Schedulability Analysis .....	127
9.6	References .....	128
<b>Part III Application and Evaluation of the SPES Modeling Framework</b>		<b>129</b>
10	Overview of the SPES Evaluation Strategy .....	131

---

10.1	SPES 2020 Evaluation Strategy.....	132
10.2	Selecting Appropriate Case Studies.....	133
10.3	Structure of the Following Chapters.....	134
10.4	References.....	135
11	Application and Evaluation in the Automation Domain.....	137
11.1	Overview: Application Domain Automation.....	138
11.2	Evaluation Strategy for the Domain.....	140
11.3	Overview of Activities and Results.....	140
11.4	Application and Evaluation of the SPES Modeling Framework.....	143
11.5	Summary.....	153
11.6	References.....	154
12	Application and Evaluation in the Automotive Domain.....	157
12.1	Overview: Application Domain Automotive.....	158
12.2	Evaluation Strategy and Correlations to the SPES Modeling Framework.....	158
12.3	Detailed Experience Reports.....	160
12.4	Summary.....	173
12.5	References.....	174
13	Application and Evaluation in the Avionics Domain.....	177
13.1	Overview: Application Domain Avionics.....	178
13.2	Evaluation Strategy and Application to the SPES Modeling Framework.....	178
13.3	Evaluation Results for each Viewpoint and Quality Aspect.....	181
13.4	Summary.....	195
13.5	References.....	196
14	Application and Evaluation in the Energy Domain.....	197
14.1	Overview: Application Domain Energy.....	198
14.2	Evaluation Strategy in the Energy Domain.....	200
14.3	Evaluation Activities and Results.....	201
14.4	Exemplary Evaluation Activities in Detail.....	204
14.5	Summary.....	212
14.6	References.....	214
14.7	Acknowledgements.....	214
15	Application and Evaluation in the Healthcare Domain.....	215
15.1	Overview: Application Domain Healthcare.....	216
15.2	Evaluation Case Study: Extended Care System.....	218
15.3	Example Evaluation Activities in Detail.....	223
15.4	Summary.....	229
15.5	References.....	230
16	Evaluation Summary.....	231
16.1	Introduction.....	232

---

16.2	Conclusions from the Evaluations .....	234
16.3	Relevance of Model-Driven Development .....	237
16.4	Summary.....	239
16.5	References .....	239
16.6	Acknowledgements.....	239
 <b>Part IV Impact of the SPES Modeling Framework</b>		<b>241</b>
17	Lessons Learned .....	243
17.1	The multitude of systems gives rise to a multitude of engineering challenges .....	244
17.2	Model-based software development is increasingly important .....	245
17.3	Integrated development is essential for the engineering of embedded systems .....	246
17.4	Interdisciplinary knowledge networks foster innovation .....	246
17.5	We have achieved a lot — but a lot more still remains to be achieved.....	247
17.6	Summary.....	248
17.7	References .....	249
18	Outlook .....	251
 <b>Appendices</b>		<b>255</b>
A	Glossary of the SPES Modeling Framework .....	257
B	Author Index .....	261
C	Project Structure.....	267
D	Members of the Innovation Alliance.....	271
E	List of Publications.....	289
F	Index.....	299

# Part I

## Starting Situation

# 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. *Increasing computing powers*  
 This development, which we already know from computer systems, is now finding its way into embedded systems.
- Future embedded systems will be more and more networked. *Networked systems*  
 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
- 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” [Schirmacher 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

*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.

#### 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.

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.*

## 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

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.

*Steadily growing complexity*

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

*Strict certification  
guidelines and safety  
requirements*

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

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