Klaus Pohl · Manfred Broy
Heinrich Daembkes
Harald Hönninger *Editors*

Advanced Model-Based Engineering of Embedded Systems

Extensions of the SPES 2020 Methodology

Springer

Advanced Model-Based Engineering of Embedded Systems

Klaus Pohl · Manfred Broy
Heinrich Daembkes · Harald Hönninger
Editors

Advanced Model-Based Engineering of Embedded Systems

Extensions of the SPES 2020 Methodology

Springer

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

Manfred Broy
Institut für Informatik,
Technische Universität München,
Garching, Bayern
Germany

Heinrich Daembkes,
Airbus Defence and Space,
Ulm,
Germany

Harald Hönninger,
Robert Bosch GmbH,
Renningen, Baden-Württemberg
Germany

ISBN 978-3-319-48002-2 ISBN 978-3-319-48003-9 (eBook)
DOI 10.1007/978-3-319-48003-9

Library of Congress Control Number: 2016955426

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Embedded systems have long become an essential part of our everyday life. They control essential features in our vehicles, 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 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 time-to-market.

Need for an integrated, model-based development approach

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, can only be applied in very specific settings, and lack an appropriate cross-domain validation in representative industrial settings.

The mission of the SPES XT project was thus to focus on the professionalization of a cross-domain, model-based development approach for embedded systems known as the SPES methodology. SPES XT is an joint research project sponsored by the German Federal Ministry of Education and Research. In SPES XT, partners from academia and industry have joined forces in order to enhance a modeling framework based on the latest state of the art in embedded systems engineering to address specific engineering challenges and to ensure that the modeling framework can be applied in embedded systems industries.

Aim of this Book

Industry challenges, principles, and application

The aim of this book is to present an overview of the SPES XT modeling framework and to demonstrate its applicability to embedded system development in various representative industry domains. The book explains the basic solution concepts of the SPES XT mod-

eling framework comprehensively and illustrates the application of these concepts in three major application domains: automation, automotive, and avionics. The book summarizes the lessons learned, outlines evaluation results, and describes how the SPES XT modeling framework can be tailored to meet domain-specific and project-specific needs.

Target Audience

Researchers, practitioners, consultants, and teachers

This book is aimed at professionals and practitioners who deal with the development of embedded systems on a daily basis. They include developers, requirements engineers, software or hardware architects, business analysts, mechatronics experts, safety engineers, testers, and certifiers. The book serves as a compendium for researchers in the field of software engineering and embedded systems, regardless of whether they are working for a research division of a company or are employed with a university or academic research institute. For teachers and consultants, the book provides a solid 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 sixteen chapters:

Status quo and industry requirements

- *Part I — Starting Situation:* This part discusses the status quo of embedded system development and model-based engineering and summarizes the key engineering challenges emerging from industrial practice. Chapter 1 gives detailed insight into the role of embedded systems and outlines the scope of the SPES XT project. Chapter 2 presents two example specifications of embedded systems from the automation and automotive industry. Later on in the book, these case examples will be the main basis for evaluations.

The SPES XT modeling framework

- *Part II — Modeling Theory:* This part introduces the backbone of the proposed model-based engineering methodology: the SPES XT modeling framework and its underlying core principles. Chapter 3 presents an overview of the SPES XT modeling framework and introduces its basic methodological concepts. Subsequently, Chapters 4 and 5 place particular emphasis on two major contributions of the SPES XT modeling framework: Chapter 4 introduces a general context modeling framework for

embedded systems development and Chapter 5 introduces concepts for the seamless integration of software and systems engineering in industrial development processes.

Solving major engineering challenges

- *Part III — Application of the SPES XT Modeling Framework:* This part describes the application of the SPES XT modeling framework in order to address the major industrial challenges identified. In particular, Chapter 6 proposes solutions to allow early validation of model-based engineering artifacts, Chapter 7 addresses the need to manage the physical context during verification activities, and Chapter 8 details the application of the framework to aid structured engineering of highly interacting system and function networks. Chapter 9 then proposes solutions to support optimized deployments of software, Chapter 10 discusses the opportunities to support modular safety assurance by applying model-based engineering techniques, and Chapter 11 shows the application of the SPES XT modeling framework for systematic variant management and strategic reuse.

Experiences and evidence from industry applications

- *Part IV — Evaluation and Technology Transfer:* This part assesses the impact of the SPES XT modeling framework. Chapter 12 summarizes the key lessons learned from exemplary applications in the automation domain. Chapter 13 discusses the value of the SPES XT modeling framework for driving technology transfer. Chapters 14 and 15 provide further evidence for the applicability of the proposed methods by means of industrial tool support as well as the applicability of the methods to industrial case examples. Chapter 16 gives a brief outlook for the future.

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 XT 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, Peter Heidl, Dr. Stefan Henkler, Dr. Ulrich Löwen, Dr. Andreas Vogelsang, and Dr. Thorsten Weyer for their relentless effort

in integrating the different project activities, for many fruitful discussions and suggestions, and for their critical reviews of project milestones. In the same way as for the preciding book "Model-based Engineering of Embedded Systems – The SPES 2020 Methodology" (Springer 2012) much of the content of this book is the 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 Dr. Thorsten Weyer (from paluno, University of Duisburg-Essen) and his team members Philipp Bohn, Marian Daun, and Bastian Tenbergen for the excellent management of the overall writing and 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 01IS12005. 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 and Jörg Nordengrün 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
Manfred Broy
Heinrich Daembkes
Harald Hönninger

Summer 2016

Table of Contents

Part I

Starting Situation

Klaus Pohl
Manfred Broy
Heinrich Daembkes
Harald Hönninger

1

Advanced Model-Based Engineering of Embedded Systems

The markets for embedded systems are characterized by high innovation pressure, steadily decreasing times to market, and the omnipresent need to reduce development costs. This trend is accompanied by the necessity of developing innovative products with greater functionality and more features that can be sold to customers. In the joint research project "Software Platform Embedded Systems XT" (SPES XT), a group of 21 partners from industry and academia came together to improve the engineering processes for embedded systems in the automation, automotive and avionic industry. In this chapter we give an introduction to the SPES XT modeling framework supporting the seamless model-based engineering of embedded systems and addressing core challenges in todays embedded systems engineering.

K. Pohl et al. (eds.), *Advanced Model-Based Engineering of Embedded Systems*,
DOI 10.1007/978-3-319-48003-9_1

1.1 Challenges in Embedded System Development

Embedded systems consist of hardware and software. The computers are often realized by microcontrollers which are typically connected to the whole system of sensors, actors, operator controls, and communication devices. Programs executed by these controllers, known as embedded software, represent an essential part of the systems because they realize the functionality of the systems. Embedded systems are an essential driver for innovation in many domains — for example, automotive, avionics, and industry automation but also in the energy domain and in healthcare, rail, and robotics applications.

Market for embedded systems

In 2012 ninety-eight percent of the microcontrollers produced worldwide are employed in embedded systems. The overall market for embedded systems has increased continually over the last 20 years and is still increasing significantly. The German Federal Association for Information Technology, Telecommunications and New Media (BITKOM) reports that the German market for embedded systems already passed the 20 billion euro mark in 2012.

Market characteristics

Furthermore, the markets for embedded systems are characterized by high innovation pressure, steadily decreasing times to market, and the omnipresent need to reduce development costs. These features are accompanied by the necessity of developing innovative products with greater functionality and more features that can be sold to customers.

Challenges

The result of these demands is that embedded systems are more complex and overall, more embedded systems are being used in contemporary products. The challenge for the industry is to develop these more complex systems with all the required features and to a high level of quality. Mastering these challenges is essential to stay competitive in the markets with respect to innovation, time to market, and cost structures. Failing to meet these goals weakens the competiveness of companies and entire application domains by limiting the innovation potential. Errors and weaknesses in the engineering of such embedded systems may have direct adverse financial consequences or even threaten human physical integrity.

1.2 The SPES Engineering Methodology

In the SPES initiative, the above-mentioned challenges were addressed by proposing a seamlessly integrated model-based, cross-domain engineering approach which addresses the specific concerns of embedded systems and their development processes. A model-based, tool-supported approach based on a solid mathematical foundation as proposed by SPES enables efficient development of embedded systems. The approach starts with an analysis of the system's context and initial customer requirements. This is followed by the gathering of system requirements and specification of the system, architecture design and implementation, and finally, verification and certification of the system.

SPES contributes a modeling framework

SPES contributes to meeting the challenges by providing a modeling framework which comprises the fundamental modeling concepts that are needed — including relationships between models, such as refinement — as well as methods for defining mappings between modeling concepts. The SPES modeling framework categorizes these artifacts into four viewpoints and allows artifacts to be specified on a dynamic model of degrees of granularity.

The viewpoints — which form the horizontal dimension — are based on the *principle of separation of concerns,* which supports a number of views of a system which each serve specific purposes. The aim is to reduce the system's complexity by considering only that information of the system under development which is relevant according to one particular development view.

Viewpoints

The SPES modeling framework distinguishes between the following four viewpoints:

- Requirements
- Functional
- Logical
- Technical

Degrees of granularity

The different degrees of granularity of systems — which constitute the vertical dimension are concerned with the decomposition of the system into subsystems following the *principle of divide and conquer*. Following this principle, the system is decomposed into smaller and supposedly less complex parts.

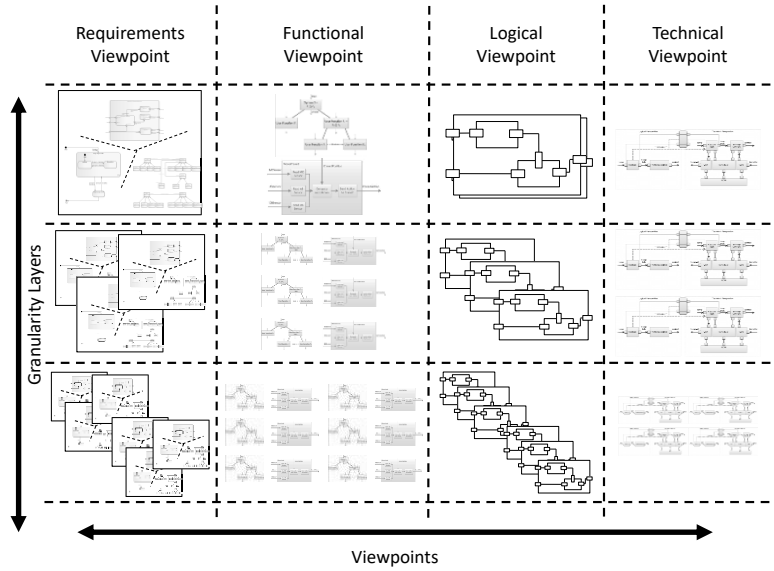

Fig. 1-1 *Structure of the SPES modeling framework*

Practical relevance

Within the SPES XT project, specific artifacts that practitioners produce in their everyday development work were identified and incorporated into the comprehensive SPES XT modeling framework, extending the SPES modeling framework. The artifacts in the framework are separated into those pertaining to the problem space and those pertaining to the solution space. SPES provides specific relationships between these artifacts to facilitate development across different degrees of granularity.

Adaptability

Note that the SPES modeling framework does not prescribe specific modeling techniques or tools. The artifacts can be created and documented using conventional modeling techniques, languages, and tools that most practitioners are already familiar with. In summary, the SPES modeling framework has laid the foundation for engineering for embedded systems in a field that is becoming increasingly demanding.

1.3 Vision and Mission of SPES XT

In the national joint research project SPES XT, a group of 21 partners from industry and academia came together to further extend the seamless methodology and analysis techniques for embedded systems. The research interests focused on six challenges in the en-

gineering of embedded systems identified by the industry partners of the project as being highly relevant. The following research areas were addressed by the SPES XT engineering challenges:

- Improving the integration between software engineering and systems engineering
- The integration of systems and ensuring adequate function in physical system contexts with a specific focus on system aspects
- Design space exploration and optimal deployment of software onto hardware components
- System quality assurance through validation at the earliest possible stage during development
- Support for management of system variants and reuse
- System qualities (e.g., safety, real time)

Project characteristics

The project was set up in a way that allowed a seamless integration of the results across the engineering challenges and across application domains, with domain-specific scenarios being discussed within the engineering challenges. The results were applied to use cases from exemplary high-tech domains (automotive, avionics, and automation).

Industrial acceptance

The SPES XT project also focused on preparing the results achieved in a way that they can be deployed in industrial practice: the first priority here was to ensure acceptance by developers and take the current domain-specific development processes into account. To achieve this goal, the project proposed a series of building blocks that can be tailored for domains, companies, and development teams.

1.4 Topics not Addressed

SPES XT concentrates on key issues in model-based systems engineering. Since the field is so broad, a number of widely known and very important topics could not be included, for example:

- Human-centric engineering
- Open context and context uncertainty
- Autonomous behavior of systems
- Collaboration of embedded systems in dynamic networks
- Uncertainty in analysis results, system behavior at runtime, and context configurations at runtime
- Runtime validation and verification

Foundation for extensions

However, the SPES projects (i.e., SPES 2020 and SPES XT) have developed a broad methodological support and thus laid a foundation for the engineering of embedded systems that can be extended in a variety of directions. A number of the topics listed above, such as open context and collaboration of embedded systems, will be addressed in our future research activities.

1.5 Key Contributions of the SPES XT Approach

As discussed above, the SPES XT project has extended the seamless methodological and technical tool approach for coping with specific challenges in the engineering process for embedded software.

We introduced a set of crosscutting topics which were relevant for all engineering challenges in order to support consistency and the practical applicability of the developed results and to make sure that the following central principles of the SPES methodology were considered in all project results:

- Seamless integration of the methodology
- Assessment of artifact quality
- Integrated tool platform and tool support
- Practical applicability

The SPES XT modeling framework is central to the project results. It defines basic terms, concepts, and theory and serves as the basis for all work in the engineering challenges. The framework has been enhanced by specific models and concepts from the engineering challenges.

Introduction of building blocks

To enable seamless methodological support on the one hand and to consider domain-specific requirements on the other hand, SPES XT has developed a set of building blocks that solve the problems of the engineering challenges. These building blocks can be used as elements that constitute domain-specific engineering processes.

Tool support

The integrated tool platform is an enhancement of the reference technology platform from the CESAR[1] project and serves as a framework for integrating academic and industrial tools. From a concept perspective, the tool platform is based on the SPES XT

[1] CESAR: Cost-efficient methods and processes for safety relevant embedded systems, ARTEMIS project

modeling framework and allows the engineering challenges to be treated holistically. The platform works in a modular way that enables the use of a suitable subset of platform-based tools which are appropriate for a given problem. These tools will immediately benefit from the additional value provided by the platform, including seamless model-based engineering facilitated by the platform-enabled integration of engineering tools.

Guidelines for practical implementation

For the transfer of the results, such as theories, models, and tools into industrial practice, we have developed guidelines for the practical implementation of the methodology. These guidelines are essential for applying the SPES methodology in specific use cases.

1.6 The Future of Embedded Systems

Embedded systems have undergone and will continue to undergo dramatic change due to the fast pace of development of digital technology. Key areas reflecting that change and the extension of embedded systems in terms of abilities and power include:

- Convergence of information and embedded systems
- Autonomous systems
- Cyber-physical systems
- Digital world
- Human-centric engineering
- Service orientation of embedded systems
- Structural and functional integration
- Integration with classical information and communication technology

These topics bring in new methodological challenges. Many of these challenges can be addressed by the techniques and principles developed in the SPES projects.

1.7 References

[Pohl et al. 2012] K. Pohl, H. Hönninger, R. Achatz, M. Broyc (Eds.): Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology. Springer, Heidelberg/New York, 2012.

[Rajan and Wahl 2013] A. Rajan, T. Wahl: CESAR – Cost-efficient Methods and Processes for Safety-relevant Embedded Systems. Springer Vienna, 2013.

Frank Houdek
Ulrich Löwen
Jan Christoph Wehrstedt

2

Running Examples

In the following chapters, we introduce advanced concepts that improve the development of embedded systems. To illustrate the concepts and increase comprehensibility, we use examples to demonstrate how the concepts can be applied.

In this chapter, we introduce two illustrative, non-trivial examples that contain typical characteristics of current embedded systems as they are developed in the three SPES application domains, namely automotive, automation, and avionics. In this chapter, we provide an overview of the examples and then in the subsequent chapters, we use specific parts or aspects of the examples to illustrate the SPES methodology.

K. Pohl et al. (eds.), *Advanced Model-Based Engineering of Embedded Systems*,
DOI 10.1007/978-3-319-48003-9_2

2.1 Introduction

Two non-trivial real-world examples

Examples are essential when illustrating new concepts. They help to explain abstract ideas as the ideas become more comprehensive. An integrated example helps us to understand how the SPES methodology addresses various aspects of the development challenge and how they interact.

Within this book, we focus on two non-trivial real-world examples to illustrate our concepts. The examples are drawn from the automation and automotive application domains. The first example is an automotive system cluster that contains two typical automotive systems, namely an adaptive exterior lighting system and a speed control system. The main functions of the exterior lighting system are:

- Turn signal (direction indicator, hazard warning light)
- Low-beam headlights (including daytime running light and cornering light)
- High-beam headlights (including automatic high beam if no oncoming vehicle is detected)

The speed control system includes the following functions:

- Cruise control and adaptive cruise control
- Distance warning
- Braking assistant and emergency brake assistant
- Speed limiter, including speed sign detection

The second example is a desalination plant with the following main functions:

- Pumping water through the processing stations
- Reverse osmosis
- Controlling water quality

Characteristics of the examples

The use of complex examples also gives an insight into the scalability of our concepts and illustrates how they can be applied to real-world problems. The examples have characteristics which are typical of today's embedded systems:

- Highly distributed functionality
- Functionality which is partly safety-critical
- Real-time demands

- Software-intense systems (i.e., software is the key element for implementing functionality)
- Combination of mechanical, mechatronic, electrical, and electronic components
- Both reactive and controlling behavior

Additionally, the automotive example covers characteristics such as:

- Many variants of one product in place (e.g., due to different product architecture, local regulations, extra equipment)
- Product evolution over time; not all engineering and documentation activities (such as the safety case) have to be repeated for the entire product

The desalination plant example from the automation domain covers characteristics such as:

- Each plant is unique and designed according to the requirements of a specific customer
- Functionality is greatly determined by software but nevertheless, the core functionality is driven by mechanics; this results in critical dependencies between the mechanics, electrics, and software which have to be managed during engineering

Given the complexity of the examples, it is not feasible to present all of the details within this chapter. Instead, here we give a brief overview of the two examples and elaborate on some aspects that are used later on to illustrate the SPES methodology in more detail.

Disclaimer

Disclaimer: The running examples presented in this chapter are inspired by or derived from real-world systems. Nevertheless, we have modified the original real-world systems so that the examples do not describe current or past real-world systems of Daimler AG or Siemens AG.

2.2 Automotive Example: Exterior Lighting and Speed Control

2.2.1 Project Setting

The project setting of this example is as follows: the automotive original equipment manufacturer (OEM) specifies the requirements of the system in text form at the granularity level *vehicle*, meaning that the system is treated as a black box. In addition, the OEM

defines the electric/electronics architecture (E/E architecture) of the vehicle — that is, the electronic control units (ECUs), the communication networks used along with gateways, and the locations of the sensors and actuators. The OEM also defines the basic configuration and extra equipment, which in turn determines optional elements in the E/E architecture and the functional requirements. For safety-related functions, the OEM also conducts a risk analysis to determine which functions are subject to ISO 26262.

Typically, a new project (e.g., for a new vehicle model) will be based on previous projects. New functions are added and existing functions are modified or removed. The automotive example addresses this facet of system evolution by providing four versions of the system specification that build on one another. Therefore, there is a need for a modular safety assessment (see Chapter 10).

Model-based systems engineering is performed primarily by the system provider. The example also covers these activities. However, due to the size of the example, the models shown in this chapter are excerpts and do not represent the complete system.

2.2.2 Functionality

The automotive system cluster contains two automotive systems, namely an exterior lighting system and a speed control system. The main motivation for using two systems is to incorporate some kind of feature interaction: the automatic, speed-dependent high beam benefits from information about the target speed determined by the adaptive cruise control.

Main functions of the exterior lighting system

The main functions of the exterior lighting system are:

- **Turn signal (direction indicator, hazard warning light):** In addition to a normal turn signal, the system also offers tip-blinking (i.e., if the pitman arm is used for less than 0.5 seconds, three flash cycles are initiated). In the USA and Canada, an activated daytime running light must be dimmed by 50% during blinking. In order to save energy, the hazard warning light switches its light-dark ratio from 1:1 to 1:2 if the ignition key is removed.
- **Low-beam headlights (including daytime running light and cornering light):** The low-beam headlights can be activated manually or — if a rain/light sensor is available — by the environmental light. The cornering light is activated during direction indication if the vehicle is slower than 10 km/h. Ambient light is activated for 30 seconds after a vehicle door has been

opened or closed. Starting the engine deactivates the ambient light.

- **High-beam headlights (including automatic high beam if no oncoming vehicle is detected):** In automatic mode, the high beam is activated if the camera does not detect an oncoming vehicle. The high-beam headlight illumination area is dependent on the vehicle speed or — if adaptive cruise control is available — on the target speed as determined by the adaptive cruise control system.

An additional optional element of the exterior lighting system is the darkness switch that suppresses exterior lighting. This is relevant for armored vehicles (e.g., police vehicles observing a suspect's house). Some functions of exterior lighting are considered to be safety-relevant: direction indicating is classified as ASIL-A, low beam as ASIL-B.

Main functions of the speed control system

The main functions of the speed control system are:

- **Cruise control and adaptive cruise control, traffic jam assistant:** Cruise control maintains the speed of the vehicle at a defined value. The target speed is adjusted via a lever near the pitman arm. The driver can adjust the speed in 1 km/h increments or — in version 2 higher (see also Tab. 2-1) — also in 10 km/h increments.

  The adaptive cruise control also takes the distance to the vehicle in front into consideration, meaning that the target speed is the lower of the defined value (set by the driver) and the speed of the vehicle in front. If the vehicle in front stops, our vehicle also stops 2 m behind the other vehicle. If the traffic jam assistant is active, our vehicle automatically starts again if the vehicle in front starts driving again.
- **Distance warning:** This function becomes active (by showing a warning symbol) if the distance to vehicle in front is lower than the speed-dependent safety distance.
- **Braking assistant and emergency brake assistant:** The braking assistant becomes active if the brake pedal is pressed beyond a certain level. If this happens, 100% braking force is applied to the wheels. The emergency brake assistant reacts to stationary and moving obstacles and gives warnings or activates the brake dependent on the time to collision.
- **Speed limiter, including speed sign detection:** The speed limiter function ensures that the vehicle speed does not exceed the maximum speed defined by the driver. In the same way as in the

cruise control system, the defined speed can be adjusted in 1 km/h increments or — in version 2 and higher — also in 10 km/h increments. With optional speed sign detection, the speed limit recognized is used as the defined maximum speed.

As mentioned above, there are four versions of the speed control system that reflect the evolution over time. The differences between the four versions are shown in Tab. 2-1.

Tab. 2-1 *Features in the four versions of the speed control system*

Version 1	Version 2	Version 3	Version 4
Cruise control	+ two-level cruise control lever	+ deceleration due to vehicle in front (= adaptive cruise control)	+ traffic jam assistant (i.e., automatic continuation after a holdup)
	Braking assistant	+ distance warning	+ emergency braking assistant
Speed limiter	+ two-level speed limiter lever	+ traffic sign detection	

Because the speed control functions have access to the brake or accelerator, they are also safety-relevant. From an engineering point of view, the evolution of the functionality over time clearly illustrates the need for a modular safety assessment to reduce the overall effort required. For instance, the difference between versions 1 and 2 within the cruise control function is only the comfort option of two increments (1 and 10 km/h) instead of one (1 km/h).

2.2.3 E/E Architecture and Allocation of Functions

As described in Section 2.2.1, in addition to the technical requirements, the OEM defines the E/E architecture of the new vehicle (see Fig. 2-1). The architecture contains both mandatory and optional elements. Some optional elements (e.g., the central display) depend on the extra equipment ordered for an individual vehicle. Other optional elements depend on the vehicle model. In a luxury model, for instance, a two-body controller architecture is used (body controller front, BC_F and body controller rear, BC_R) whereas in a compact vehicle, only one body controller is used (body controller common, BC_C) and the overhead control panel (OCP) is connected to the infotainment gateway (IGW) instead of the BC_F. The E/E architecture also supports electric drive vehicles.

The functional breakdown of a vehicle function and the allocation of the individual functional contribution to individual ECUs is a key engineering challenge for an OEM. Fig. 2-2 presents one function allocation for the turn indicator functionality.

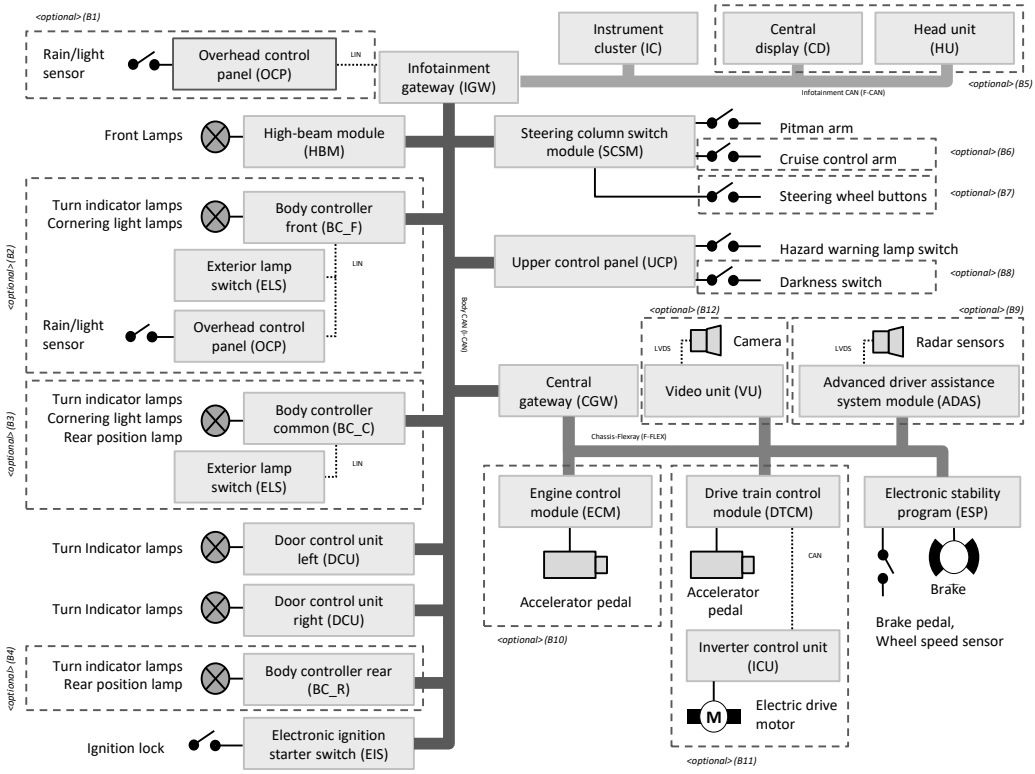

Fig. 2-1 *E/E architecture with optional and mandatory elements*

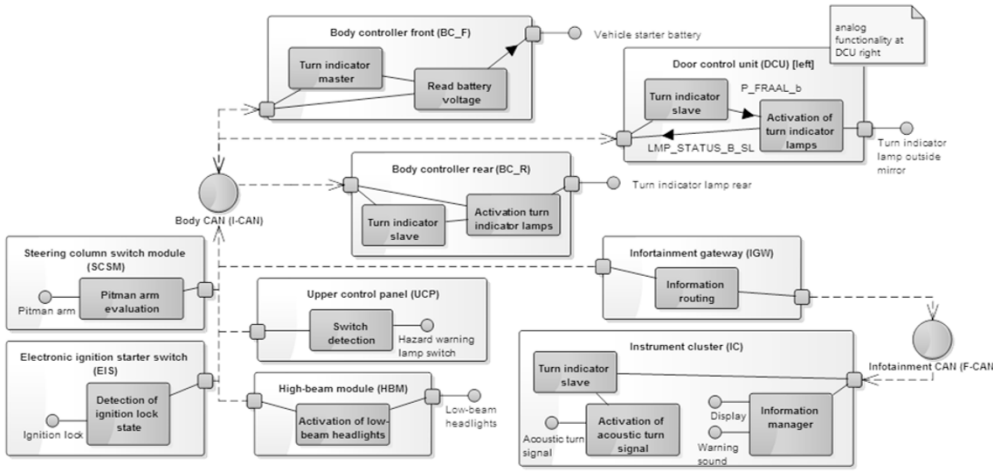

Fig. 2-2 *Function allocation for a turn indicator*

2.2.4 Variability

There are various places where the system contains variability:

- Optional equipment: Optional equipment such as (adaptive) cruise control is available dependent on the buyer's choice.
- E/E architecture: Depending on the underlying E/E architecture, a component has to communicate with different peers. As the functionality expands (e.g., cruise control vs. adaptive cruise control) the allocation to ECUs might change — for instance, from the engine control module (ECM) to the radar unit (RU).
- Country-specific regulations: Different behaviors may have to be implemented dependent on local regulations. In our example, we have the fictitious regulations that (1) in the USA and Canada, while the turn indicator is active the daytime running light must be dimmed by 50%, and (2) that auto-repeat is not permitted when setting the cruise control speed.
- Backward compatibility: Starting with version 2, the cruise control and speed limiter can be adjusted in 1 km/h and 10 km/h increments. This requires that the steering column switch module (SCSM) must physically support two increments — that is, the SCSM must support two switching thresholds. Over the production life cycle, it is common to replace a component with its successor (e.g., for cost reasons) while other components remain unchanged. This means that a new ECM that implements

the cruise control functionality has to support both one-increment and two-increment SCSMs.

- Configuration of behavior: The increment size of 1 and 10 km/h defined in version 2 of the cruise control specification becomes more flexible in version 3. Here, the increments can be configured.

The treatment of variability information in the development process is addressed in Chapter 11.

2.2.5 Engineering Challenges

The main engineering challenges for this example are as follows:

- An optimal E/E architecture and distribution of the functionality has to be identified, see Chapter 9.
- While functionality or architecture is evolving over time, there is a need to handle safety assessments in a modular way so that (small) changes do not require a full safety reassessment, see Chapter 10.
- The product (e.g., ECU) is typically used in many different contexts and has to provide different functionality depending on its environment. Therefore, the engineering process has to explicitly support variability in its many facets, see Chapter 11.

2.3 Automation Example: Desalination Plant

Desalination plants are designed to remove salt from seawater in order to produce drinking water. Desalination typically uses reverse osmosis — a filtration method which requires water to be pumped through membranes at high pressure. This example is based on a real Siemens automation project executed in Spain. The plant has a capacity of 14,000 m^3/day and a high level of automation.

Overview of a desalination plant configuration

Fig. 2-3 shows a schematic overview of this type of desalination plant configuration. Seawater is collected through four beach wells along the coast which are dug into the seashore. From the beach wells, the salt water is pumped through pipelines to the seawater tank, where it is collected, stored, and pre-treated with various chemicals for stabilization and biochemical cleansing before desalination can take place. The subsequent desalination steps include cartridge filtering and repeated treatments until drinking water is obtained in the high-pressure section. Throughout the desalination

process, the water quality is consistently monitored and adjusted to ensure that the desalination takes place at optimal efficiency and pollutants and hazardous materials can be removed.

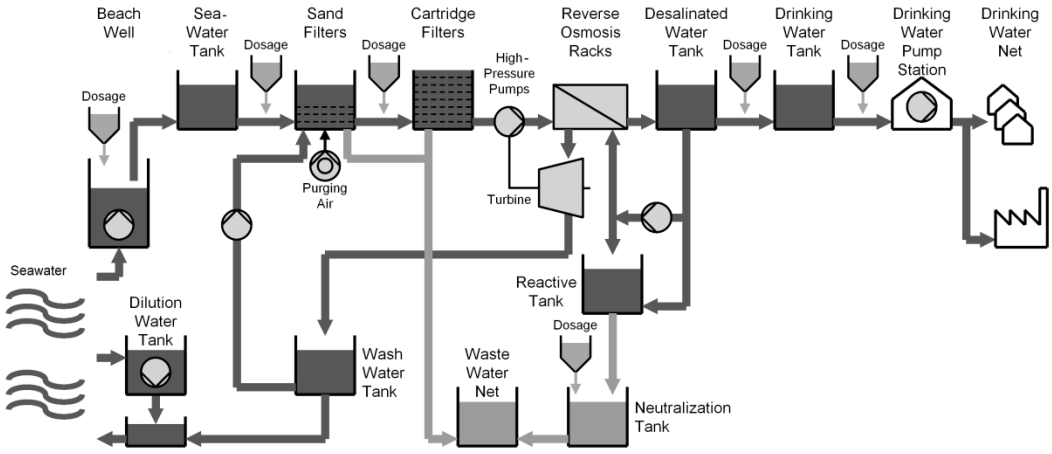

Fig. 2-3 *Configuration of a desalination plant*

To illustrate the engineering challenges, we will focus on the beach well. A beach well has the following general functions:

- Collect seawater through subsurface intakes dug into the seashore
- Filter seawater through natural sand layers
- Pump water from the subsurface intake collection tank to the seawater tank

Adjust the flow into the seawater tank

Automation industry uses piping and installation diagrams

In the automation industry, process plant configurations are designed based on piping and installation diagrams (P&ID). Fig. 2-4 shows the P&ID of one of the four beach wells of our example, outlining the principle parts of the beach wells. Each beach well is equipped with a pump to transport the water collected, a discharge valve through which water is connected to the seawater tank, a bypass control valve which can be used to adjust the flow rate delivered by the pump to avoid damage — for example, due to cavitation — and a hypochlorite valve through which chemicals are added to disinfect the water in order to prevent biological growth in the subsequent filter process.