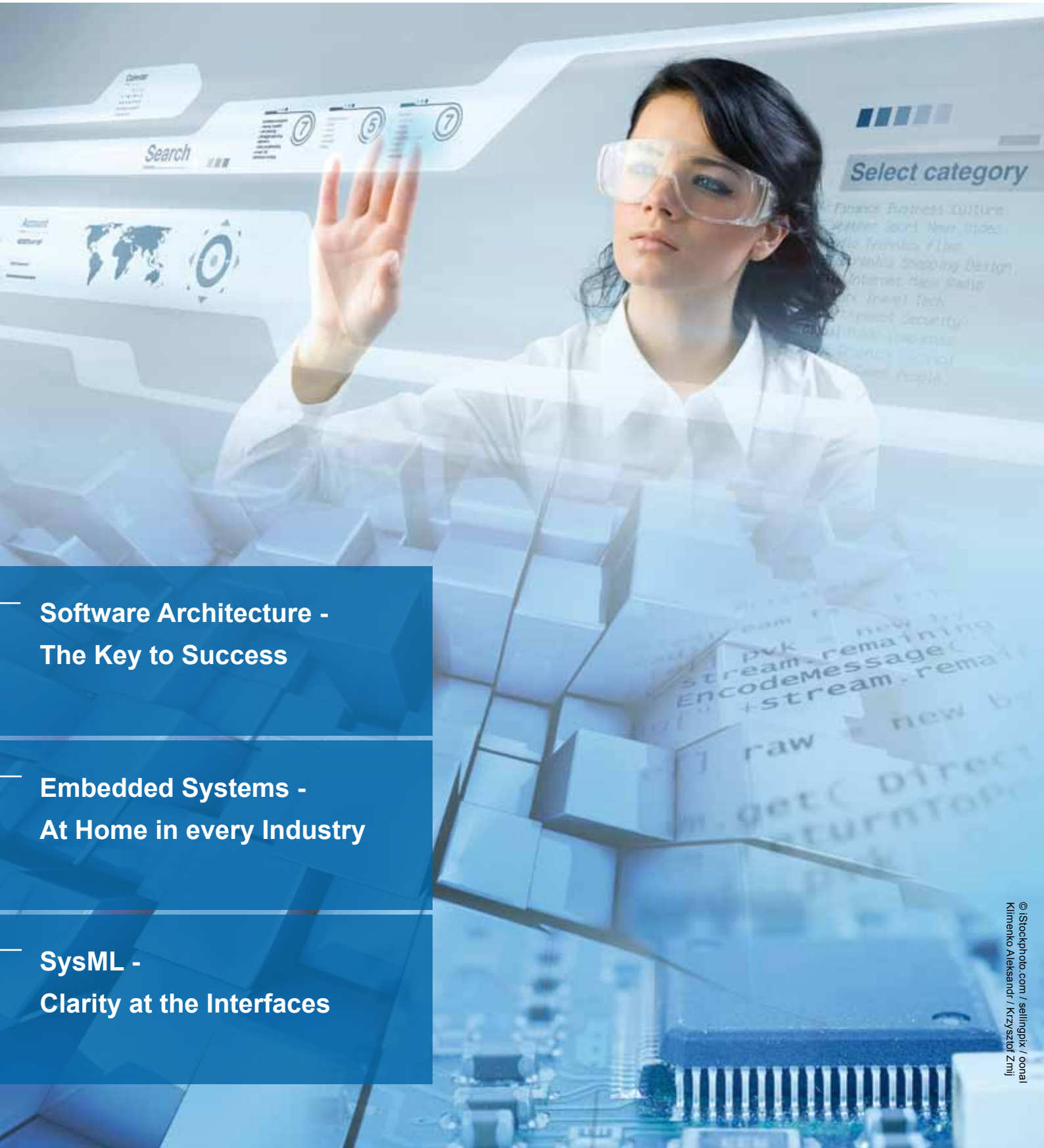
No. 4 / December 2011

Insight

berner&mattner
optimizing your development

Industry



Software Architecture -
The Key to Success

Embedded Systems -
At Home in every Industry

SysML -
Clarity at the Interfaces

PREFACE

Dear Reader,

At Berner & Mattner, you often read and hear the term "embedded systems", which has gained even more importance with the integration into the parent group Assystem. For this reason, this issue of "Insight Industry" is all about "embedded systems" and the different perspectives on these.

Due to the versatility and the requirements variety, a general characterization of embedded systems is quite difficult. Berner & Mattner uses the term for embedding computing power into technical products in order to make them more valuable and powerful, meaning computer systems that are embedded in devices, systems and machines to run special applications. Their software and hardware often build functional units that are exclusively designed to fulfill the defined task. These systems have to be real-time capable under the constraint of minimized software development efforts - all under cost and reliability aspects.

Typically, such systems can be found in aviation and aerospace technology, railway systems, automation and automotive technology – precisely those market segments that are directly addressed by Berner & Mattner providing solutions for optimizing development tasks.

Immerse yourself in the world of ECUs by reading the thrilling articles of my colleagues and discover some interesting bits and pieces of news. Enjoy your journey of discovery,

Your

Dr. Christian Hock
Head of Industry Division

IMPRINT

Publisher:
Berner & Mattner Systemtechnik GmbH
Erwin-von-Kreibig-Str. 3
D-80807 Munich
Phone +49 (0) 89 608090-0
Fax +49 (0) 89 6098182
www.berner-mattner.com
marketing@berner-mattner.com

Editorship and Layout:
Simone Horra, Dr. Christian Hock
with thanks to the authors of the papers

TABLE OF CONTENTS

Software Architecture for Embedded Systems

From implicit to explicit architecture

Documented software architecture has clear benefits. Products are more adaptable and future-proof. In addition, the knowledge of interfaces, communication links and resources of individual components facilitates their adaptation to new hardware environments and functional requirements. In the field of enterprise software, defining software architecture is standard today, but this is often neglected with respect to embedded software.

It is worthwhile to involve a software architect not only in new developments but also in the modernization of existing embedded solutions to make improvements future-proof.

Software architecture is the structured arrangement of system components, indicating the communication relationships between the components and their mapping on hardware or software resources. The ideal solution is early and ongoing support during a project by a software architect being independent from the individual developer team. A number of tasks can be assigned to this architect. In many areas, however, only moderation is needed. A central task of embedded software architecture is to consider not only the functional software requirements but to also capture non-functional requirements against the background of a life cycle prediction. Among these non-functional requirements are issues as complex as maintainability, modifiability, scalability, portability, connectivity, usability, reliability, safety, performance, efficiency or testability.

Justified technology decisions

Although a well-established team of developers is beneficial with respect to communication, it may be limited in its solution finding capability by the experience from previous or similar projects. Proven approaches to finding a solution are then often repeated without reflection, not considering new and potentially more appropriate and particularly flexible technologies.

The tasks of a software architecture team or an outside consultant may, for example, consist in evaluating new technologies and assessing their suitability for projects as well as providing

new tool sets. Components required in the future have to be identified. Involving an independent architect has often proven beneficial for proposing optional technologies for individual components.

It is all about discussing design decisions, defining interfaces and developing coding conventions. Under the moderation of a software architect, these decisions are made in a more conscious and better documented way. This is the only way to keep the context of a system reproducible for developers who need to work on it later and know nothing of the implicit assumptions and practices within the team of the system creators.

Implicit architecture

Unfortunately, the daily practice of embedded development usually gives a different picture. Time and cost constraints often prevent developers from concentrating their thoughts and efforts on systematic software architecture. In fact, each software system has a certain architecture, even though it may not explicitly have been modeled. If these structures are the accidental result of the components' development dynamics, it does not impact the functionality of the software for the time being. Problems occur only later during maintenance, scalability and porting. In retrospect, it has fatal consequences to quickly decide on a programming language or an interface under time and cost pressures in the development process, often without any documentation. The communication between the single, not clearly separated components may have been solved only superficially, instead of paying attention to minimizing potential friction already when choosing the technologies and languages. If a software project is not seen in terms of its entire life cycle, embedded projects run the risk of causing unintended dependencies on development hardware. Adaptation to new hardware generations can thus be very expensive or even impossible.

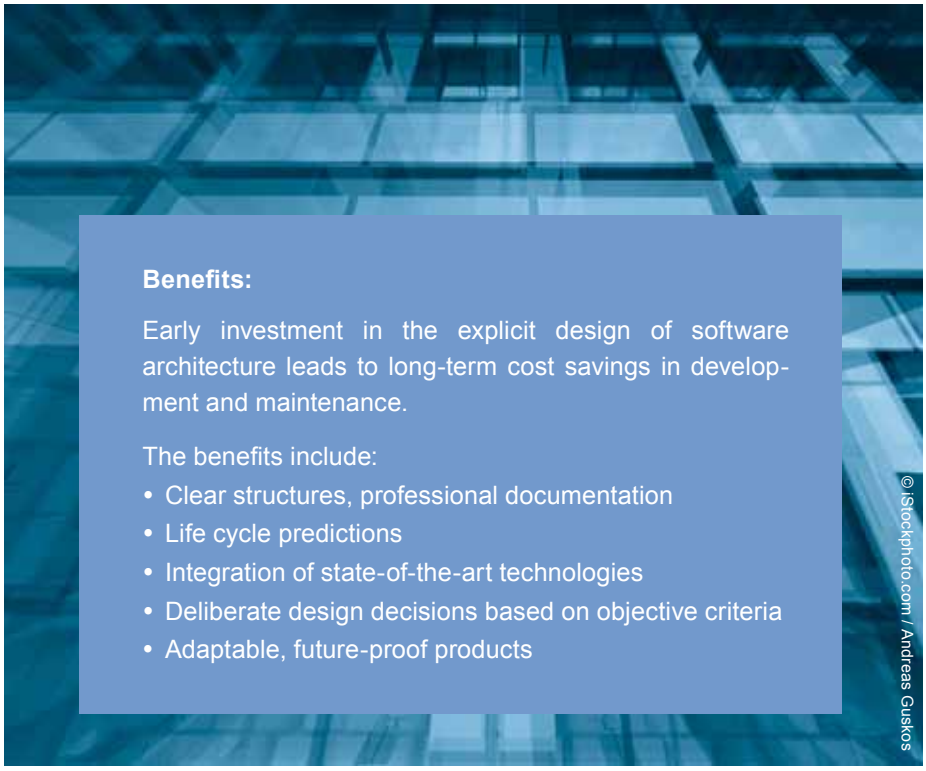
Subsequent analysis

Clearly structured and documented software architecture is missed only when issues and difficulties occur during further development which might be necessary. Often, essential parts of the software are to be maintained but connected e. g. to a new user interface. Sometimes, the controller using the software specially designed for him is not available any more and the successor requires a different design. Functions have to be extended, merged, or adapted to external systems that have been changed. Maintenance or further development of such a system is an unpredictable cost factor for the product manager, especially when the core of the original development team is no longer available.

Specialists like Berner & Mattner offer assistance in such situations and analyze legacy systems, redocument existing software architecture and identify possible migration and change paths. As external architecture consultants, they can analyze the architecture´s problems irrespective of the software's history and propose solutions. Based on such an analysis, the expenditure for maintenance and further development can be evaluated in a better way.

Setting and breaking rules

The subject is particularly exciting with respect to certified embedded systems. A common challenge is to have the software or significant parts of it



Benefits:

Early investment in the explicit design of software architecture leads to long-term cost savings in development and maintenance.

The benefits include:

- Clear structures, professional documentation
- Life cycle predictions
- Integration of state-of-the-art technologies
- Deliberate design decisions based on objective criteria
- Adaptable, future-proof products

For more information please visit: www.berner-mattner.com/en/sw-architecture

certified in order to comply with a SIL (Safety Integrity Level). Of course, nobody wants to go through an IEC 61508 certification process again, if this can be avoided. Fortunately, often only a few relevant parts of the system are classified as safety-critical. The skill of the architecture consultant is to find the locations in the software where cuts can be placed between the safety-critical parts and these components and interfaces that can easily be changed. Common design patterns help him perform this task, such as layers or "pipes and filters". Ironically, the rules generated this way - the derived architecture for the actual development project - have to be broken again in order to maintain the components that have just been certified. But the big difference to the "rule violations" from the time the original software was written is that they are now well documented.

Software architecture as a process

It would be a misunderstanding to see software architecture as a finalized document that is touched only rarely again. It should be read and maintained every time the software is touched. Without this living document, architecture violations are often not recognized. Agreements on architecture, coding conventions, interfaces, etc. are thus becoming binding for the developer. The client receives an embedded system, whose maintenance can be handed over to a new team without requiring a lot of explanation or training. The specialists who have created it and who are now relieved from this can dedicate themselves to other tasks. This means that software architecture is never finished! The objective of the design methods to be applied by the developers, such as Object Oriented Analysis and Design (OOAD), are clear structures and the separation of responsibilities: This, obviously, is architecture!

Clear software architecture helps to better understand the relations within a system, preventing failures and reducing the efforts when "renovating" embedded systems. Given the long life cycles of many embedded systems, the subsequent analysis and documentation is worth the effort if the software architecture has been neglected during the embedded development. Otherwise, unpredictable delays happen again and again when maintenance or further development encounter undocumented peculiarities in the system. An external software architect provides valuable assistance with respect to teams with old habits and to knowledge building.

Software architecture is the structured arrangement of system components, indicating the communication relationships between the components and their mapping on hardware or software resources.

Most failures created by human error occur where we do not fully understand what we are working on. Explicit architecture contributes greatly to making the causal chains of complex systems transparent. With this knowledge, necessary changes will be evaluated a lot faster and implemented reliably. Early investment in the explicit design of software architecture leads to long-term cost savings in development and maintenance.

Electromobility

Embedded software in charging infrastructure

Berner & Mattner has extensive know-how in the field of electromobility, including the car interface, the hybrid powertrain as well as advanced driver assistance and telematics systems, amongst others.

Electromobility is on everyone's lips and the widespread use of electric vehicles is only just beginning. What we do know is that the charging infrastructure is a decisive building block for the commercial success of electric engines. Crucial for the wide acceptance are the intuitive operation of the charging stations and the embedding of the charging infrastructure in a holistic concept providing the user with a real added value, e. g. when parking.

Task

The interfaces between the charging infrastructure and external systems are defined on the basis of the requirements analysis. The model-based approach of Berner & Mattner offers the advantage of using executable models, facilitating the verification of processes together with the customer at very early development stages. These models also facilitate the important comprehensive communication with the vehicle.

Intuitive operation of the charging systems

Two factors are crucial during the software development for the charging infrastructure:

- It is essential to have an intuitive workflow-oriented operating procedure with a modern user interface.
- The surface design should be exchangeable without affecting the functionality in order to fulfill the requirements of the charging station provider.

Central software: early load tests are crucial

The challenges during the central software development are different in nature: The basis is an extensible software architecture applying modern internet-based technology which supports the dynamic binding of adapters to external systems without disrupting the running system as a whole.

The development of automated test environments through hardware simulation (SiL - Software in the Loop) allows for the parallel development of software and hardware as well as the early execution of load tests to prove that the center will meet the expected number of charging cycles in the future.

Modern technology at Berner & Mattner

Berner & Mattner relies on modern and proven technologies for the development of user interfaces such as WPF and XAML, achieving clearly defined interfaces between design and functionality. They can be transferred to production software by external users in real time. Reliable continuous integration platforms with automated module, integration and system tests (e. g. VMWare, Jenkins, nUnit) are used for the development and tests. Not only test cases within the business logic are covered, but also automated GUI tests. This is possible by considering testability already during the definition of the software architecture (design for testability) and the strict adherence to design patterns like MVVM (Model-View-View Model).

Engine Controls / Applications

Requirements engineering

Hybrid systems, new emission and fuel consumption limits and high-availability systems integration – the leading motor manufacturers are facing numerous challenges. Reliably networked electronics and modular software are essential solution components and decide on the positioning of competing systems. Berner & Mattner supports several global market leaders in their system- and software engineering processes, starting with the requirements analysis.

Systems engineering means defining requirements, describing system structures, fixing internal and external interfaces, all in our customers´ language, in order to understand each other unambiguously and to develop the right system and corresponding test cases.

The integration of engines into commercial and military equipment and systems requires compliance with comprehensive industry and country-specific standards. Berner & Mattner analyzes these standards, often volumes amounting to several thousands of pages, and reduces the contents to relevant prioritized key statements:

- The basic requirements are specified in a specific and legible way.
- Long passages are reduced to essential contents.
- Often, regulations contain seemingly harmless short texts but with hidden implementation consequences.
  These complications will systematically be detected.

In the subsequent gap analysis, we identify the major deviations in the electronic and software systems of our customers in terms of the standard requirements.

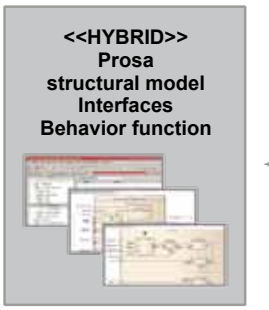
Requirements and functional safety analysis

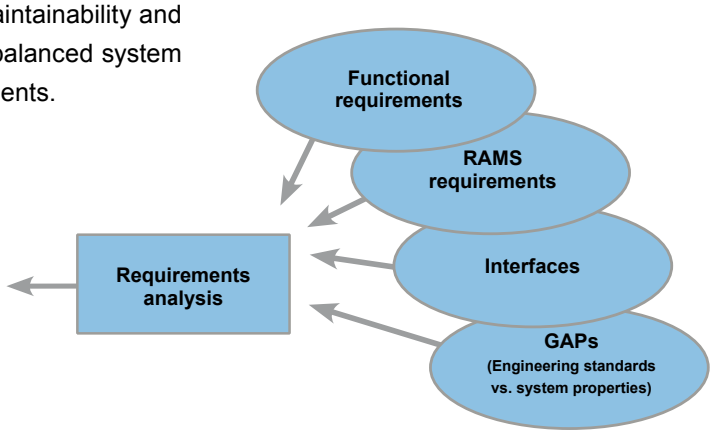
Berner & Mattner uses best industry practices for requirements engineering, e. g. use cases, scenarios, structured analysis, SysML, UML, in order to describe essential functions such as controllers, turbo-charger modes, diagnosis, start-up behavior, emergency operations, etc. The analysis of external and internal interfaces and the execution of functional safety analysis are of particular importance. Berner & Mattner accomplishes a holistic perspective with the RAMS-model – Reliability, Availability, Maintainability and Safety – and derives balanced system and software requirements.

HYBRID specifications

Hybrid is not only a 'hot topic' but exceptionally efficient, both for drives and system specification. Our experts are utilizing a combination of text-based and model-based description for high-quality specifications.

Our customers benefit from the clarity combined with good comprehensibility and systematic traceability for test and design at the same time.





For more information please visit: www.berner-mattner.com/en/motorcontrols

Embedded Systems in Space

Breaking new ground with spin-ins from other industries

Berner & Mattner´s experience in embedded software from other industries can provide significant contributions to the development of future on-board satellite software.

Not only in the exploration of the earth, our solar system and the cosmological contexts in the universe, the European Space Agency ESA sets out on new paths. Learned lessons from other industries, so-called "spin-ins", play an increasingly important part in further developing the technologies used.

Electronics & embedded software

The relevance of electronics and embedded software in complex satellite systems is growing steadily and is increasingly important for the success of a mission. However, the historically evolved practice to develop electronics and the respective software for planned missions from scratch is no longer appropriate. A large part of the SW functionalities of comparable (sub-) systems converges, allowing room for more efficiency, both in the development of such software and in quality assurance.

Spin-in for the space industry

Against the background of the ever-growing share of electronics and embedded software, the space industry can increasingly benefit from the experience and best practices gained in other industries.

In the development of highly complex satellite systems, the European Space Agency ESA can for example gain great advantage through the experience made in the automotive industry or in the area of other transport systems. In current research, knowledge gained in the development of AUTOSAR-standards or in cross-vendor use of communication standards is compared with the knowledge of past and current satellite developments. Initial results of such studies caused ESA to launch an initiative aiming at standardizing software architecture for on-board software.

Breaking new grounds with our experience

Berner & Mattner has many years of experience in developing embedded systems for several industries and foremost in safety-critical environments. As a result, our company can also give crucial impulses and spin-ins for the upcoming changes in the technological foundation of further developments in space. This means that even "rocket science" is a field open to our ideas! We regularly raise them in current discussions, thus supporting to create a solid foundation for future-oriented developments for space science.

Distributed Embedded User Interfaces

Consistent HMIs for defence electronics

Major software projects, in particular in the defence sector, are often implemented across sites and countries. The adjustment and development of a consistent operating logic for the front end of embedded controllers is challenging. Berner & Mattner provides a proven architecture and technology platform for such tasks, supporting customers in their international workshare.

Berner & Mattner has successfully used the "Eclipse Rich Client Platform" as a basis for a distributed HMI development. With this, numerous embedded front ends have already been developed successfully.

During the development of complex defence systems, a proven way to make the complexity manageable is the architectural segmentation of the components into individual layers, with HMI as the user interface.

For complex HMIs, it is appropriate to use this layer view also internally. The distributed development of the user interface in more than one team makes an abstraction of the actual operating logic from communication mechanisms even imperative.

The basis is a common communication layer, specifically tailored to system-wide communication mechanisms but generic in terms of the data transported.

This common communication layer provides decisive advantages during development:

- The individual components (though often very complex) can be processed in terms of their specific control logic and data flows.

Another key concept of Berner & Mattner are the reusable, graphical modules (widgets), the technological basis enabling the developer team to provide a largely consistent look and feel. Input fields with consistent validation are good examples, but complex tables, too, are standardized in this way. This integrated approach makes it possible to let distributed development teams implement the technical components of the user interface, while safeguarding the important usability and maintenance requirements.

Key competences

- Prototypical and iterative-incremental HMI development processes
- Systematic automated tests and continuous integration
- Military standards (MIL-2525, MIL-1477, MIL-1472)
- Profound and long-standing experience in using the Eclipse RCP

DB Netz

High-quality interface specifications with SysML modeling

Benefits of modeling with SysML:

- More precise requirements
- Better coverage
- Less failures
- Easier to understand

Berner & Mattner has been using SysML in many projects with very good results. One of these projects is NeuPro by DB Netz.

The development, specification and standardization of interfaces between subsystems are required in many areas where embedded systems are employed. Within the project NeuPro of DB Netz, Berner & Mattner has designed a model-based approach to develop high-quality interface specifications with SysML. During the three years of development, the methodology has matured and proven to be very suitable for complex projects with a huge number of participants. It is based on modern systems engineering concepts such as System of Systems (SoS) and part of an overall process for the development of complex systems with SysML.

When employing embedded systems, many industries would like to define and introduce standardized interfaces between subsystems with the objective of creating a market, promoting competition, reducing costs and increasing quality. The AUTOSAR-standard in the automotive sector is an example of such an initiative. In the railway industry and during the project NeuPro, Berner & Mattner has developed a model-based method for DB Netz to develop interface specifications between interlockings and field elements (such as the signaling) through modeling with SysML.

Approach

The approach consists of two phases:

- The modeling of the domain level
- The modeling of the technical level

The domain level provides a functional, logical and abstract view of the requirements, irrespective of particular solution concepts. Solution-focused requirements, such as physical, electrical or software-related descriptions, are covered at technical level. Tasks carried out at domain level are referred to as the analysis phase in systems engineering.

The technical level uses a technical solution concept to implement domain level requirements. In systems engineering, this phase is referred to as the design phase.

Activities at domain level

At the beginning, it has to be determined which of the subsystems of the overall signaling system are involved in the interface communication (in the following referred to as interface end points) and which of the subsystems are important regarding the interface to be specified. This is referred to as

the interface context (interface environment). SysML applies a block definition diagram for this purpose. It shows the static structure of elements (in SysML: blocks) and their relation to one other (associations).

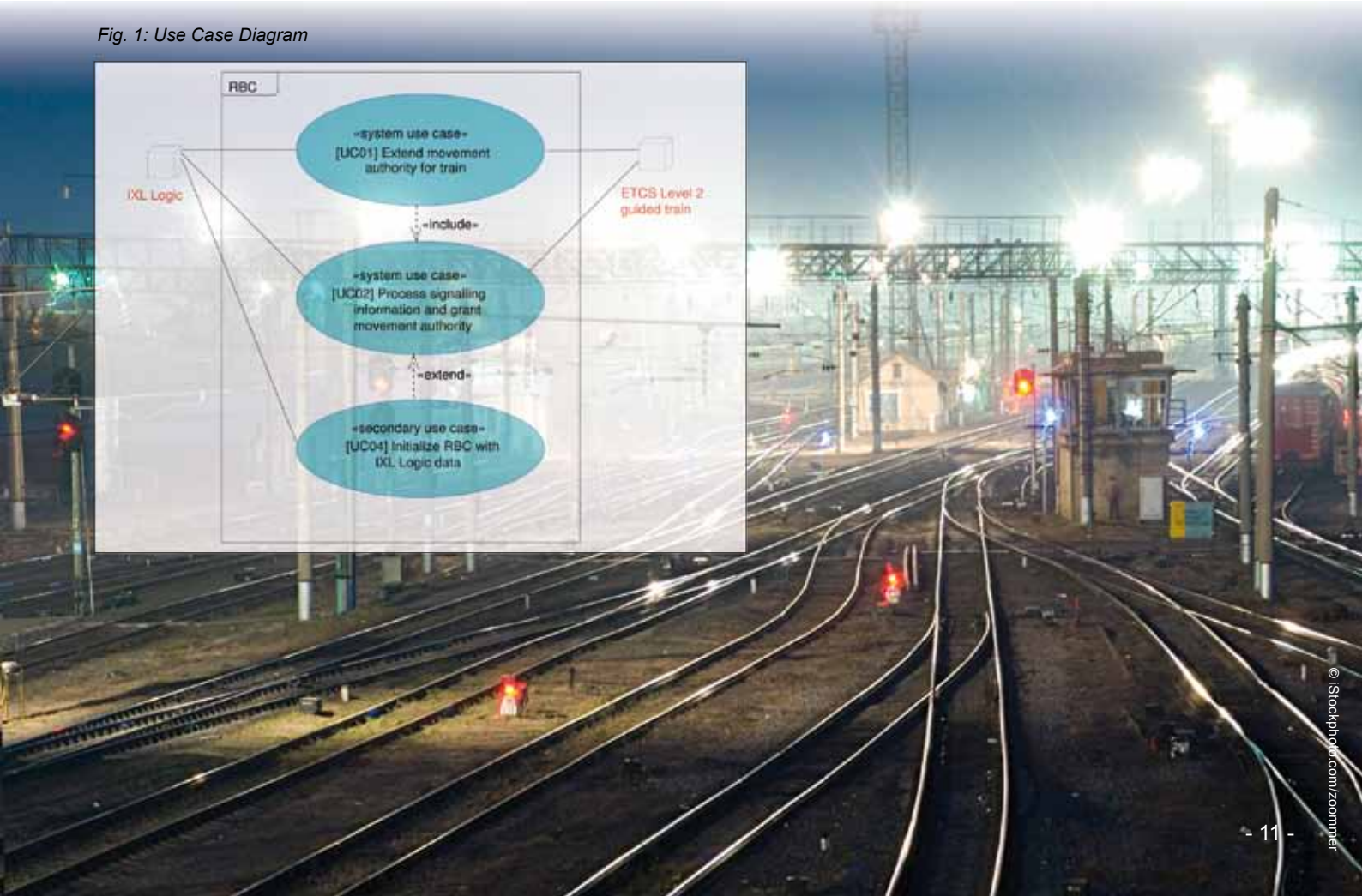
Subsequently, the interface definition determines which functionality is executed on which subsystem of the interface context and which of these functionalities require communication via the interface. A use case analysis is performed in order to determine these functionalities. In SysML, use cases reflect the functionality of a subsystem as services which a subsystem offers to another subsystem or a person (actors). After defining the interface context, the interface-relevant use cases have to be identified and modeled in a use case diagram (figure 1). A use case consists of a sequence of actions executed in changing order by the subsystem and the actor who is applying the use case. The activity diagram of SysML models this sequence of alternating actions between actor and subsystem. It is determined in detail which subfunctionality is executed on which subsystem and in what sequence.

Once the use case sequences have been worked out and the functional distribution among the subsystems has been fixed, the next step is the functional interface specification, based on sequence diagrams (figure 2). The necessary communication via the interface with functional information (commands and messages) is determined for standard workflows (successful process flow of a use case without faults) of the developed activity diagrams. Sequence diagrams are suitable for visualizing the interaction of several selected communication scenarios, but not for completely describing all processes. Therefore, a new diagram type comes into play: Statecharts for subsystems of interface end points now also cover and model all exceptional cases. In this way, the interface description is very complete at domain level. Lastly, statecharts offer execution and simulation options in order to enable further optimization and requirement testing.

Activities at technical level

The final step is the description of the specific technical implementation. At first, the ISO/OSI layers are described. If industry protocols and standards are applied such as Ethernet and TCP/IP, it is sufficient to refer to their specification. Finally, the technical data telegrams are defined, including the description of their bytes and bits as well as their value ranges and meanings.

Fig. 1: Use Case Diagram



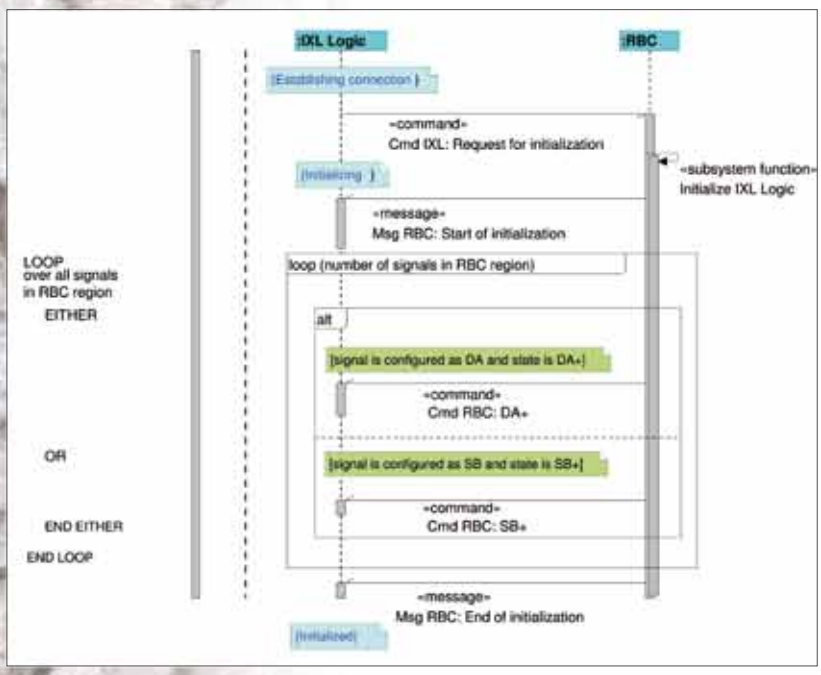
SysML process of Berner & Mattner

The process for developing interface specifications is part of a comprehensive process. It is developed by Berner & Mattner for modeling and developing railway systems with SysML.

The entire process

- is based on the procedure models SYSMOD (Systems Modeling Process) and OOSEM (Object-Oriented Systems Engineering Method)
- implements the concept System of Systems (SoS)
- defines a model structure (implemented with SysML packages)
- supports the modeling of variants
- integrates the phases of the RAMS life cycle
- supports three presentation levels of railway requirements (operational, domain and technical level)
- is easier to understand due to the limitation of the number of SysML model elements used, compulsory modeling guidelines, integration and preference of railway terms as well as presentation in German language as far as possible.

Fig. 2: Sequence Diagram



Software Standards for Satellite Applications

Design guide for efficient implementation

Control software used in satellites or to control the satellite from the ground must meet highest requirements. Accuracy and reliability are key criteria to ensure a successful execution of the satellite mission. Therefore, the software development has to meet highest standards and the development process is laid out precisely.

The design guide as a key reference manual is a reliable tool to meet the high requirements concerning the development of satellite software.

The European Space Agency (ESA) has defined the requirements concerning the development of its projects in ECSS-standards addressing different aspects of development in various sub-documents. For software development, for example, not only the standards for software engineering and software quality assurance have to be considered but also the general standards for project management or systems engineering.

The developer of software for satellite applications thus needs to know many different types of documents in order to take all requirements into account during development. In addition, a specific project tailoring has to be performed to define the specific measures that have to be implemented depending on the criticality of the application.
In order to provide the developers of satellite software with a reliable tool for the application of the required processes, it is appropriate to develop a manual defining the concrete requirements and measures. This has several advantages:

- Key reference for all relevant requirements of the development process
- Consistent requirements for process tailoring
- Clarification of unclear or ambiguous wording in standards

This procedure, for example, was implemented in the Galileo project. With a distributed development at various locations and with different companies involved, a design guide also helps to efficiently implement a consistent approach.

Berner & Mattner supports its customers in the introduction and implementation of sophisticated development standards by means of developing process manuals, defining and introducing methods and implementing them into appropriate tools. Here, standards from different industries are applied for developing safety-critical systems. In addition, we provide a relief to our customers from industry and science by accepting responsibility for the efficient and reliable transfer of product assurance and project management tasks.

Industrial Embedded Systems

Berner & Mattner at embedded world 2012

embedded world 2012
Exhibition & Conference
... it´s a smarter world

In early February, we will present our range of services in the field of "INDUSTRIAL EMBEDDED SYSTEMS" at embedded world 2012, showing how we combine proven best practices in a very useful way and how we apply our cross-industry synergies even more efficiently.

Together with our colleagues of the French Assystem Group, we will present our joint engineering and consulting expertise with respect to the life cycle of electronic ECUs and control systems – from specification, development and test automation to series production support.

We would be pleased if you marked February 28 to March 1, 2012, in your calendar and visited us at stand 326 in hall 5. With the help of several exhibits we will present you with innovative customized solutions and interesting business models.

News Headlines

>> Next Events

Berner & Mattner presents its services at Congress "Embedded Software Engineering" (December 6 - 8, 2011) in Sindelfingen. Come and see us!

Info: www.ese-kongress.de/english

Berner & Mattner will again take part in the embedded world in Nuremberg next year (February 28 - March 1, 2012). We look forward to seeing you at our stand 326 in hall 5.

Info: www.embedded-world.de/en

>> Presentation at "ESE Congress"

Dr. Michael Sturm, Berner & Mattner, will give a lecture on the topic "Architecture Life Cycle Optimization with Embedded Design Patterns" in Sindelfingen on December 7, 2011.

Info: www.berner-mattner.com/ese2011 (German only)

>> Presentation at "ERTS 2012"

At the "Embedded Real Time Software and Systems" in Toulouse (February 1 - 3, 2012), Berner & Mattner will give a lecture on the topic "Requirements and Test Case Tracing".

Info: www.erts2012.org

>> Technical Paper "Function-Oriented Specification as a Contract between Principal and Manufacturer"

This paper presents a specification technique for the description of system interfaces which is now being used successfully at Berner & Mattner.

Info: www.berner-mattner.com/fachartikel (German only)

>> Technical Paper "Safety Standards as a Means of Improving Quality"

In this article, Berner & Mattner presents an approach for the model-based development of safety-critical systems derived from the V-model of the CENELEC standards family.

Info: www.berner-mattner.com/fachartikel (German only)

>> Technical Paper "Integration of Functional Safety in the Development Process"

Berner & Mattner presents a new methodology for the tool-based combination of safety activities with traditional system development and early implementation of safety requirements into the system design.

Info: www.berner-mattner.com/en/technicalpapers

>> News from the Product Corner

Version 3.3 of the simulation and test platform MESSINA is available as of now. In addition to a GUI redesign, system speed and timing accuracy have been improved.

To further strengthen our teams of the Industry Division in Munich (m/f) we are looking for:

Software Architects

Project Managers Software Development

Software Developers Embedded SW

Software Engineers .NET Industrial Automation

Software Developers Java - 3D Visualization with OpenGL

Development Engineers Safety-Critical Systems

We offer graduates and career changers (m/f), in particular, an intensive and individualized training program. Berner & Mattner provides attractive conditions, individual development plans, team support, freedom to build your personal career and the opportunity to take on responsible tasks.

No suitable position for you? If you are experienced in technical software development and have skills in railway, automotive, aerospace, defence or industrial automation, we look forward to receiving your application. We will review your records and contact you as soon as we have an appropriate position to fill.

For detailed information about the company and the individual positions, please visit **http://www.berner-mattner.com**. We look forward to receiving your convincing application. Please use our online application form at **http://www.berner-mattner.com/bewerben** (German).

www.berner-mattner.com/de/jobs

Ms Anja Ranft
Human Resources Manager
Erwin-von-Kreibig-Str. 3
D-80807 Munich
Phone (089) 60 80 90-201
Fax (089) 609 81 82
Anja.Ranft@berner-mattner.com

berner & mattner
optimizing your development

berner & mattner
Systemtechnik GmbH

The previous issues of *Insight* Industry (issues 1 and 2 are only available on our German website) can be downloaded at:

www.berner-mattner.com/en/download-center/newsletters

May 2011

>> Member of Assystem - Berner & Mattner becomes International

>> Distributed Engineering - Quick, Safe and Efficient Development

>> Software Development - Reliability across Industries

October 2010

>> Systems Engineering for Various Applications

>> Partnerships with Industry and Science

>> Tools for Software Development

April 2010

>> Model-based System Specification in Aerospace

>> Virtual Build and Test Environments in Product Development

>> Innovative Solutions with Human Machine Interfaces