AUTHORS

Dr. rer. nat. Dipl.-Inf. Thorsten Weyer
led until December 2020 a Research Group at Paluno – the Ruhr Institute for Software Technology/ Workgroup Software Systems Engineering of the University of Duisburg-Essen and works currently as a System Architect at Schaeffler Automotive in Bühl (Germany).

Dipl.-Ing. (FH) Marcel Goger
is Specialist in Process Management Systems Engineering at Schaeffler in Nuremberg (Germany).

Dr.-Ing. Walter Koch
is Head of Advanced R&D Engineering at Schaeffler in Nuremberg (Germany).

Birgit Kremer, M. A.
is responsible for Public Relations at Paluno – the Ruhr Institute for Software Technology of the University of Duisburg-Essen (Germany).

Implementation Strategy for Seamless Model-based Systems Engineering

Systems engineering based on models allows a better understanding and management of complex system architectures. However, many organizations find it difficult to switch to model-based development. The University of Duisburg-Essen and Schaeffler demonstrate how it can be achieved with a sound methodology, an implementation concept, and appropriate employee training.

© gorodenkoff | Getty Images

1 MOTIVATION

Currently developed vehicles are based on a highly networked system architecture whose software functions are constantly being further developed. Development is commonly distributed across organizations, disciplines and domains. In order to generate a common understanding of the system, a large number of documents are usually created at different levels of abstraction. Starting with documents describing the goals of end customers, OEMs, suppliers and so on, documents are written to describe functions. This in turn is built upon by documents with a detailed specification of the components of a system. These are often linked to each other, resulting in a complex web of documents that is difficult to keep consistent over the lifecycle of a development. One solution to this problem is Model-Based Systems Engineering (MBSE). In the Software Systems Engineering Group at the Software Technology Institute Paluno of the University of Duisburg-Essen, a concept for the introduction of MBSE was developed and evaluated using the example of a concrete application at Schaeffler.

2 MASTERING THE DOCUMENT FLOOD

In MBSE, a central system model is at the center of development activities and, depending on its design, can be used by all the disciplines involved. It stores all information about the development project, from the requirements to the interface description. The resulting collaboration can take place at an early stage of the project and can be extended to subsequent projects. Among other things, this results in the following advantages:
– better division of labor
– mastery of increasing complexity
– reusability of development artifacts
– mastery of variability and variants
– automation in the generation of models, code and tests
– automation of quality assurance activities
– clear communication through precise modeling language and models.

3 CHALLENGES

Although many companies are aware of the benefits of early model-based development, MBSE has not yet arrived in their day-to-day business. One reason for this is that employees do not yet have sufficient skills: a lack of access to a well-founded methodology as well as integrated tools to apply a methodology consistently throughout the entire development process. On the other hand, the organization itself is a challenge. In order to benefit from MBSE in the long term and to overcome the major obstacles, **FIGURE 1**, a considerable amount of additional work has to be done, especially in the early development phases. Although frontloading as a strategy for reducing costly verification and validation tasks at hardware-in-the-loop or vehicle level is accepted and established, model-based architecture and requirements development with MBSE starts even earlier in the product development process. This additional effort has to be supported by the management and not least by the employees. The latter have to change their usual work processes, which will only be successful if they are convinced of the benefits and a personal reduction in workload.

4 TOOLBOX OF METHODS

In the last ten years, the Software Platform Embedded Systems (SPES) modeling framework has been developed in the context of large-scale research projects – a comprehensive methodology for end-to-end model-based system and software development of (collaborative) embedded systems or Cyber-Physical Systems (CPS) [1-3]. The core of this methodology is to consider a system from different viewpoints and to develop models of the system that are as formal and consistent as possible. The viewpoints are tailored to the roles in the product development process (requirements engineer, function developer, system architect, etc.) and focus on the sections of the system to be developed that are relevant for the respective role. By creating and linking the models of individual viewpoints, a consistent database of interconnected artifacts is generated.

Selected architectural components are considered as independent (sub)systems for which, in turn, models are created using the

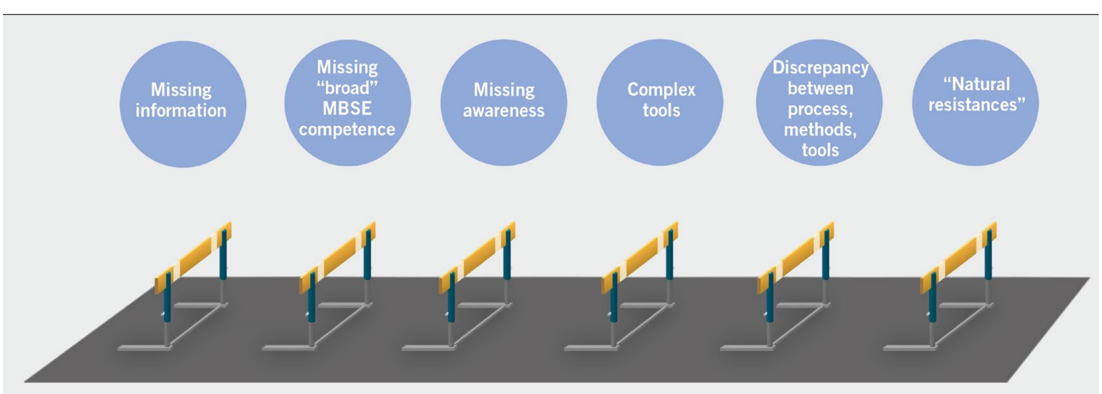

FIGURE 1 Obstacles to the transformation into MBSE (© Schaeffler)

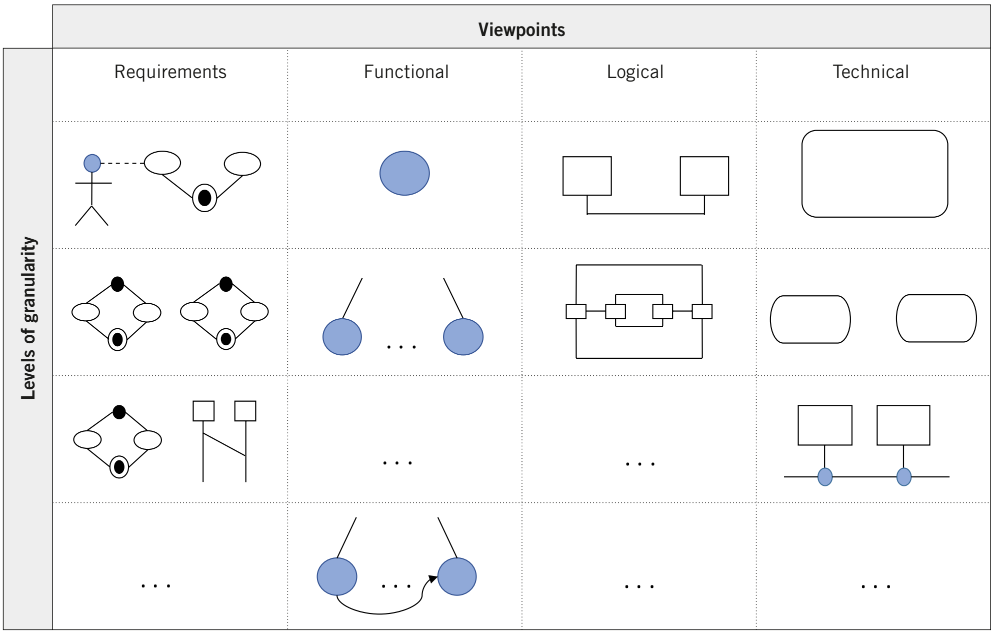

FIGURE 2 SPES matrix of exemplary viewpoints and corresponding levels of granularity; pictograms [1] for possible modeling (© Paluno)

SPES approach. This results in several levels of granularity at which a system and its subsystems are described. The so-called SPES matrix, **FIGURE 2**, summarizes the viewpoints at the granularity levels. The respective viewpoints contain a comprehensive and consistent set of concrete modeling techniques and activities. For each cell in the matrix, the SPES methodology proposes a set of models to describe the necessary information. For example, the analysis and specification activity from the requirements viewpoint deals with context, goal, scenario, and requirements modeling.

5 CONTINUING EDUCATION

In order to establish the SPES methodology in practice, comprehensive multimedia learning materials for step-by-step introduction were developed in the SPEDiT technology transfer project. The concepts and materials were evaluated in pilot projects at companies in various industries. The learning material is based on the typical roles and their necessary competence profiles. For each role, such as that of a system architect, it was determined at which learning level the corresponding knowledge is required. The learn-

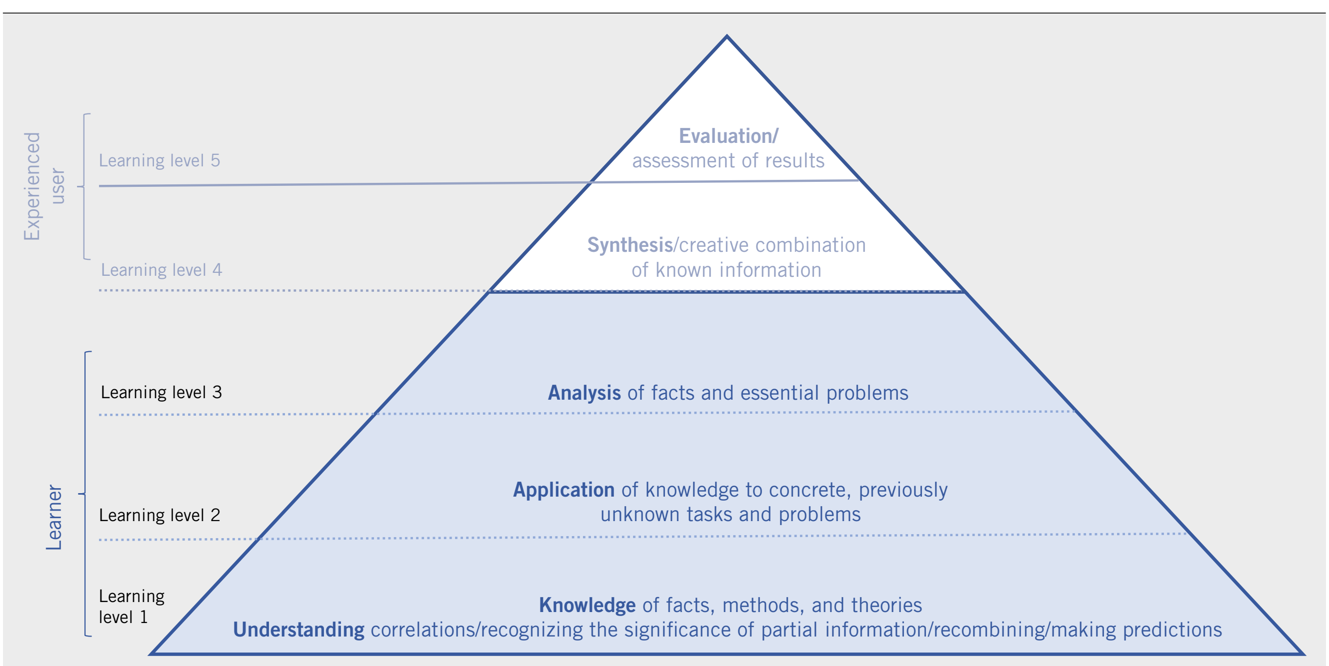

FIGURE 3 Learning objective levels and corresponding learning objectives (© Paluno)

ing levels considered in the development of the learning material are shown in **FIGURE 3**. The learning material was developed up to learning objective level 3, at which learners are able to analyze more complex issues in a structured way and express them in the form of a model. **TABLE 1** shows the five realized training modules, which are thematically divided into learning contents and to which the respective learning objective levels are assigned. The training module system architect, for example, consists of four subject areas. The learning content for requirements modeling and deployment is at learning objective level 1; it only needs to be understood at this level. The contents of the topics functional architecture and logical architecture are on learning objective level 2 and 3, respectively. After successful completion, learners must be able to uncover complex functional relationships and, for example, specify functional architectures with the help of models. Each module takes approximately 60 h to complete. The training content can be integrated into various online learning platforms such as Moodle.

The learning material is based on the concept of blended learning, meaning learners acquire knowledge independently and check their own learning progress with the help of control questions and modeling exercises. The basic content is taught in instructional videos. The learning control for learning objective levels 2 and 3 is carried out, for example, via modeling tasks that can be processed by the learner online with the help of modeling tools. Accompanying the individual learning, the pilot projects also used web seminars in which learners could exchange information with experts or ask questions about the learning content.

Learning content	Learning level
Module 1: System architect	
Requirements modeling	L1
Functional architecture	L2–3
Logical architecture	L2–3
Deployment	L1
Modul 2: System requirements engineer	
Context and scenario modeling	L2
Scenario modeling	L2
Specification/analysis of requirements	L2–3
Requirements refinement	L2–3
Module 3: Test engineer	
Requirements modeling	L1–2
Modeling test case specifications	L2–3
Test execution	L3
Simulation	L3
Module 4: Software architect	
Context and goal modeling	L2
Scenario modeling	L2
Specification/analysis of requirements	L2–3
Functional architecture	L2–3
Logical architecture	L2–3
Module 5: Software developer	
Requirements modeling	L1
Specification of software components	L2
Formal analysis	L3
Simulation	L3
Scheduling synthesis	L3

TABLE 1 Learning modules, associated subject areas and learning levels (© Paluno)

6 PILOT CONCEPT

For the step-by-step introduction of the SPES methodology in the pilot project at Schaeffler, a concept was developed, **FIGURE 4**, which allows a continuous success control and steering of the introduction of individual modeling methods. The development of an integrated parking brake for the Schaeffler e-axis was selected as a pilot project for the application of MBSE. It was suitable as a pilot application because the mode of operation is quite simple. At the same time, the project had the necessary depth of development to evaluate the SPES methods: The development ranged from the mechatronic system to the software code.

The pilot concept provides at the beginning an assessment of the existing capabilities with regard to MBSE. This phase is supported by the application of an MBSE maturity model. This firstly serves to assess the current maturity of a development organization with regard to MBSE, and secondly to develop a strategy focused on improvement goals for how the respective organization can be further developed in a stepwise and compatible manner.

FIGURE 5 shows a section of the maturity model for the areas of requirements, system functions, architecture and testing. Each of these engineering functions is subdivided into focus areas, for each of which specific maturity levels are defined. Based on the respective characterization, it is possible to comprehensibly assess whether a development organization has already reached this maturity level or which measures should be taken to improve the organization in the corresponding focus area. The assessment with the MBSE maturity model usually takes only 2 to 3 h, but provides valuable insights for defining improvement goals for MBSE. **FIGURE 5** shows a part of the results of a maturity analysis. The cells colored in gray represent maturity levels already achieved by the organization in the individual focus areas. The cells containing letters (A, B, C ... F) define capabilities in the respective focus area that must be met in order for the organization to reach the appropriate maturity level in that area.

Approximately 15 Schaeffler employees were involved in the pilot project of the integrated parking brake, with varying degrees of intensity depending on their role. The employees first learned the new methods through self-study and applied the modeling techniques they had learned to the development of the parking brake. Every two weeks, the employees met for coaching sessions with MBSE experts from the Paluno institute and other project partners. After several cycles, it was checked whether the goals had been achieved and new measures were prepared. During the two-year pilot, the majority of the modeling methods were introduced and put through their paces – from model-based requirements engineering to system and software architecture through to model-based testing. While the participants were quite skeptical at the beginning of the project, positive attitudes prevailed throughout at the end of the pilot. Model-based development highlighted decisive advantages: Ambiguities come up for discussion and clarification at an early stage – across disciplines and unequivocally.

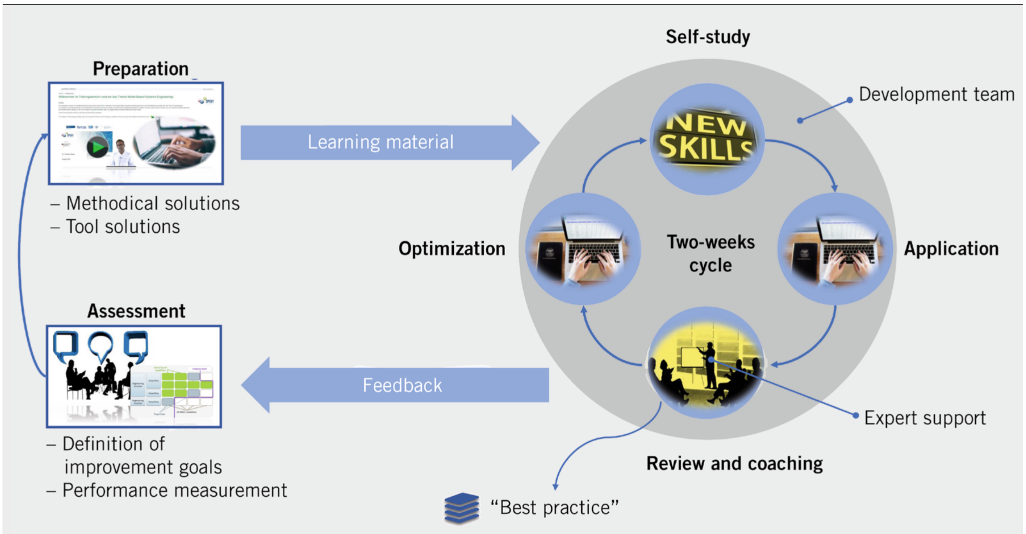

FIGURE 4 Schaeffler pilot concept (© Schaeffler)

		Maturity level										
Engineering functions	**Focus areas**	0	1	2	3	4	5	6	7	8	9	10
Requirements	Context modeling			A	B	C		D		E		
	Goal modeling			A	B	C		D				
	Scenario modeling			A		B	C					
	Solution-oriented requirements		A			B	C	D		E		
System functions	System function modeling			A	B			C			D	
	System function specification					B	C	D				E
	Event chain modeling					A	B	C		D		
	Mode modeling			A		B		C				
Architecture	Logical component modeling					A		B	C			
	Logical architecture modeling			A	B	C		D		E	F	
	System testing		A			B		C		D		

Quick wins

FIGURE 5 Example of an assessment result (blue) and possible next steps for implementation of MBSE (© Paluno)

7 CONCLUSION AND OUTLOOK

To date, more than 200 employees at Schaeffler have completed the training courses, which were built on the basis of the SPEDiT training material. The training measures have been evaluated positively across the board. While external MBSE experts still accompanied the introduction in the pilot project, the company now has its own experts who are driving forward the company-specific characteristics of the MBSE solutions and supporting the introduction of the methodology in other divisions. Awareness of model-based system development and thus recognition of its importance has continuously increased in the company, resulting in demand for further MBSE implementation. At Schaeffler, the SPES methods, training materials and, above all, the experience gained from the pilot are a valuable basis for further optimizing processes, methods and tools for product development.

THANKS

This paper is part of the results of the research projects Spedit, Crest,SPES 2020 und SPES XT, which were funded by the German Federal Ministry of Education and Research (BMBF). The working group of Prof. Dr. Klaus Pohl of the Ruhr Institute for Software Technology Paluno at the University of Duisburg-Essen worked on this project with partners from industry and science. The authors would like to thank all partners and colleagues, especially Torsten Bandyszak, Jennifer Brings, and Dr. Marian Daun.

REFERENCES

[1] Pohl, K.; Hönninger, H.; Achatz, R.; Broy, M. (ed.): Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology. Berlin/Heidelberg: Springer, 2012
[2] Pohl, K.; Broy, M.; Daembkes, H.; Hönninger, H. (ed.): Advanced Model-Based Engineering of Embedded Systems: Extensions of the SPES 2020 Methodology. Heidelberg: Springer, 2016
[3] Böhm, W.; Broy, M.; Klein, C.; Pohl, K.; Rumpe, B.; Schröck, S. (ed.): Model-Based Engineering of Collaborative Embedded Systems: Extensions of the SPES Methodology. Heidelberg: Springer, 2021