



**HAL**  
open science

# The Evolution of the V-Model: From VDI 2206 to a System Engineering Based Approach for Developing Cybertronic Systems

Martin Eigner, Thomas Dickopf, Hristo Apostolov

## ► To cite this version:

Martin Eigner, Thomas Dickopf, Hristo Apostolov. The Evolution of the V-Model: From VDI 2206 to a System Engineering Based Approach for Developing Cybertronic Systems. 14th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2017, Seville, Spain. pp.382-393, 10.1007/978-3-319-72905-3\_34 . hal-01764166

**HAL Id: hal-01764166**

**<https://inria.hal.science/hal-01764166v1>**

Submitted on 11 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# The evolution of the V-model: From VDI 2206 to a System Engineering based Approach for Developing Cybertronic Systems

Martin Eigner, Thomas Dickopf, Hristo Apostolov

Institute of Virtual Product Engineering, University of Kaiserslautern,  
Gottlieb-Daimler-Straße, Building 44  
67663 Kaiserslautern, Germany  
{Eigner, Thomas.Dickopf, Apostolov}@mv.uni-kl.de

**Abstract.** By talking about complex systems, systems engineering is always named as the only way out for the enhancement of system understanding and the reduction of system complexity in the design process. After an identification of the essential aspects and concepts for pursuing systems engineering, this paper shows how well these key factors are integrated in today's methodologies for developing mechatronic and cybertronic systems. The content of this paper is based especially on current and previous research activities on the field of model-based development at the Institute of Virtual Product Engineering.

**Keywords:** Systems Engineering, Cybertronic Systems, System Design, Design Methodologies, System Lifecycle Management, Complex Systems

## 1 Introduction and Motivation

Contemporary and future technological products are multi-disciplinary systems developed by multiple engineering disciplines with a significant level of complexity [1]. By talking about complex systems, we talk about systems with a large number of diverse and highly interconnected elements. These systems are characterized by dynamic system boundaries and cross-linkages between their elements [23]. Systems like these, which have the capabilities to communicate with each other, collect and distribute information or are able to autonomously adapt their behavior based on information available across different systems, are termed as Cyber Physical Systems (CPS) [2] [3] or Cybertronic Systems (CTS) [4] [5]. In order to handle the rising complexity of today's innovative and multi-disciplinary products, it is necessary to rethink and refine current design methodologies, processes, IT solutions as well as the entire enterprise organization. This paper shows an approach how essential aspects and concepts of systems engineering can be integrated in the development process of mechatronic systems in a model-based way to support the reduction of system complexity in the design process of mechatronic and cybertronic systems. In order to analyze today's design processes regarding to the incorporation of systems engineering aspects, chapter two gives a brief

overview of design methodologies and extensions in the field of mechatronic and cybertronic systems. While chapter three identifies essential aspects and basic ideas each systems engineering process should include, chapter four compares how well the different approaches from chapter two implements them. This paper ends with chapter five, which summarizes the results after demonstrating how the introduced approaches complement each other.

## **2 Development of Mechatronic and Cybertronic Systems**

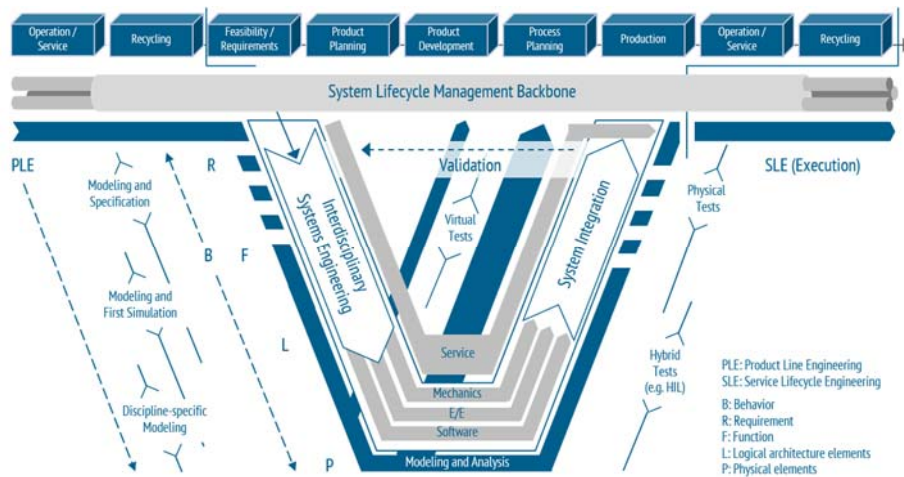
### **2.1 VDI 2206 – Design Methodology for Mechatronic Systems**

During the last decades, dozens of methodical approaches for the development of new products or the further development of existing products have emerged in the field of mechatronic systems development [6]. The best known representative of these methodologies is the guideline VDI 2206 [7]. As a supplement to the guidelines VDI 2221 (systematic approach to the development and design of technical systems and products) and VDI 2422 (systematical development of devices controlled by microelectronics), VDI 2206 is intended to describe the methods of developing mechatronic systems. The objective of this guideline is to provide methodological support for a cross-domain development especially in the early phase of development, concentrating on system design. As a whole, the guideline consists of three essential elements: a general problem-solving cycle as a micro-cycle, the V-model as a macro-cycle, and predefined process modules for recurrent working steps. In the description of the micro-cycle, the guideline VDI 2206 refers to the problem-solving cycle used in systems engineering (see [8]). In general, the micro-cycle supports the work on predictable and consequently plannable subtasks as well as the solution process of suddenly occurring and unforeseeable problems. The macro-cycle guides along the logical sequence of important sub-steps in the development of mechatronic systems. Based on ideas from software development, the generic procedure is implemented along the V-model (see [9] [10]). Some of these sub-steps, which keep recurring when designing mechatronic systems, are described in the guideline in a more concrete way. The process module system design is essential for the interdisciplinary development. Its aim is to establish a cross-domain system architecture. This architecture describes the main operating characteristics of the future product. Therefore, the overall function of a system is broken down into main sub functions, which are assigned to suitable operating principles or solution elements. [8]

### **2.2 The MVPE Model for Multidisciplinary Product Development**

The MVPE Model is an extension of the VDI guideline 2206, more precisely an extension of the macro-cycle of the guideline, which has been developed at the Institute of Virtual Product Engineering (University of Kaiserslautern, Germany) in the last years [6] [11] [12] [13]. The extensions focus on two essential points: the support of the left “wing” of the V-model by methods from model-based systems engineering and on the seamless integration and management of data from the entire product lifecycle by a System Lifecycle Backbone. With regard to the left “wing” of the V-model, *Eigner*

*et al.* identifies three levels of modeling: modeling and system specification, modeling and first simulation, and discipline specific modeling (see figure 1) [11]. On the specification level, the system is described by qualitative models, which include the system requirements as well as the functional and logical system structure. These models are descriptive and cannot be simulated. For an early system description, [11] recommend the use of modeling languages like SysML. The second level, modeling and first simulation, focuses on the integration of quantitative aspects by the creation and use of multidisciplinary simulation models (in e.g. Matlab or Modelica). On the last level, the system is modeled more precisely in a discipline-specific way. These models include discipline-specific aspects like e.g. concrete geometry representations and built by specific CAx tools. Parallel to these overlapping levels, the information artifacts or model elements are differentiated in requirements (R), functions (F), logical architecture elements (L) and physical parts (P), which are modeled in languages using authoring tools along the three levels of modeling [13].



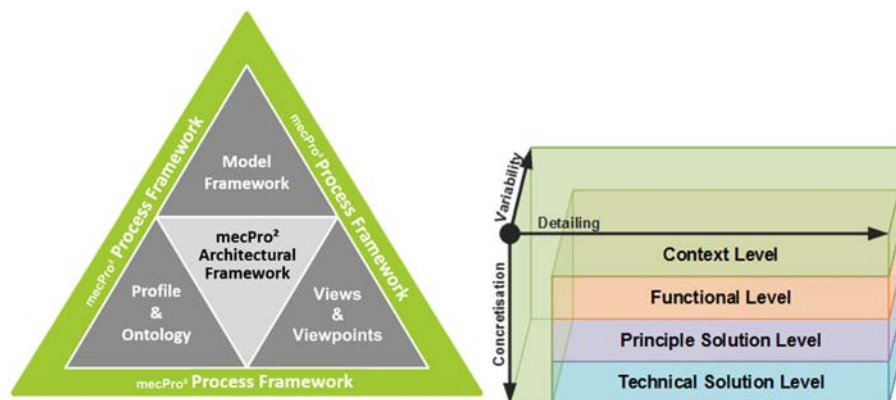
**Fig. 1.** The MVPE-Model (after [13])

*Gilz* developed a SysML-based interdisciplinary approach for the creation of a model-based system architecture based on a functional and logical breakdown in the early phase [12]. This approach, the SE-VPE method, guarantees both “horizontal” and “vertical” traceability along the different model elements (R-F-L-P) and is as well construed for a transfer of this elements into a System Lifecycle Management (SysLM) solution. Similar to Product Lifecycle Management (PLM), SysLM [14] [15] [16] is a general information management solution extending PLM to the early development phase and all disciplines along the lifecycle including services [13].

### 2.3 The mecPro<sup>2</sup> Architectural Framework

With the fourth industrial revolution in engineering, mechatronic systems enhanced to cybertronic systems. To handle the complexity of such innovative, interdisciplinary,

and interconnected products and their production systems, a rethinking of current design methodologies, processes, IT solutions, and the entire enterprise organization is needed [17]. The German research project mecPro<sup>2</sup> (Model-based Engineering of Products and Production Systems) seized on these requirements and created a concept to increase the efficiency of development projects in the field of Cybertronic Systems by using Model-Based Systems Engineering [18]. One result of the project is the mecPro<sup>2</sup> Architectural Framework. Integrated in the mecPro<sup>2</sup> Process Framework (another result of the research project [5]) the Architectural Framework is an interdisciplinary, model-based approach to describe a system during the phase of system design supported by the modeling language SysML. The mecPro<sup>2</sup> Model Framework, as an essential part of the architectural framework, forms the foundation for the description of the technical system in the early phase (see figure 2). It implements basic ideas of various development methodologies in the fields of mechatronic, mechanic, electric/electronic, software and systems engineering [17] [18], especially the RFLP approach from the MVPE model [6] [11], the viewpoints of the SPES Modeling Framework [19], the consideration of principle solutions [20] [21], and the subdivision in requirement and solution space including the three axes of detailing, variability and concretisation derived from the so-called Munich Model of Product Concretisation [21].



**Fig. 2.** The mecPro<sup>2</sup> Architectural Framework and its Model Framework [18]

As shown in figure 2, the description of the system consists of four levels with increasing solution concretization. On the context level the system is described as a black box with its interfaces. The focus of this level is the translation of natural language requirements into a system model. This includes the distinction of the system of interest in regard to its context-based environment as well as a detailed description of the expected system behavior. On the functional level, non-redundant and solution neutral system functions are identified based on the defined system behavior. The result of this level is a hierarchical and structural depiction of the system functionality including all material, energy and signal flows. On the principle solution level, the technical aspects, which realize the desired function, are considered. Therefore, principle solution alternatives should be systematically identified, analyzed and evaluated to make an optimal

selection with respect to the requirements. The evaluation and selection should be made in two stages: first with respect of the degree of fulfilment of a function and second with respect of the degree of fulfilment of possible principle solution structures, which are based on the functional structure of the level before. On the technical solution level, the maximum concretization of a solution, for which an organizational unit is responsible for, is reached. The concept to identify the final system structure is similar to the one of the principle solution level. Thereby, solution components will be identified, which realize the system functions by applying the chosen principle solution. [17] [18]

### 3 Systems Engineering

All methodologies and approaches of chapter two include or are based on concepts or aspects from the field of systems engineering. In general, especially if technical products become more and more complex, systems engineering e.g. model-based systems engineering looks like a common concept to solve the problem [8] [22] [23]. *INCOSE* describes System Engineering as an interdisciplinary approach for the realization of successful systems by considering the whole problem [24]. In the context of problem solving, *Haberfellner et al.* describes systems engineering as the methodical factor that helps to synchronize other problem solving factors to find the best solution [23]. Therefore, the system design in system engineering is based on two fundamental concepts. Systems thinking as a mindset, that enables a better understanding and redesign of complex systems, and a procedure model based on basic principles and components to support the development and realization of a solution by subdividing them into understandable sub-steps [8] [23].

Systems thinking supports holistic thinking within interdependencies as well as the differentiation and the structuring of the system. Thereby, it contains the essential terms as well as exemplary approaches for the description and illustration of complex object, without unallowed prohibited simplifications. *Crawley et al.* defined four tasks, which base on the essential features of a system to aid people in practicing systems thinking [22]. The first task is to identify the system, its form, and its function. Each system has form and function, whereby the form is the instrument of function. In the most cases, the primary function of a system is clear. The second task is to identify the entities of a system, their form and functions as well as the system boundary and context of use. System entities are, in general, also systems, which have a form and a function. The system of interest itself could be an entity of a larger system. Important to know is, what is part of the system under development and what is interacting with the system in its context? Based on this, task three helps to identify what are relationships among the entities of the system and at the boundary. Each link between the system entities as well as links to entities outside the system have a formal and functional character. The fourth task is to identify the emergent properties of the system based on the functions of the entities and on their functional interactions. It is the synergy that gives the system its power, because through the interaction between the entities a new function or characteristic arise, that is greater than the sum of the functionalities of its parts [22]. *Haberfellner et al.* clarify in their approach, that system thinking can be characterized by

different perspectives of the system [23]. Therefore, it is essential to describe the system by models, which specify a specific problem of the reality in an abstract and simplified way. The identified perspectives are environmental, impact, structure, and hierarchical oriented. The environmental orientated perspective serves to identify factors, which influence the system or get influenced by it. The impact oriented perspective considers the system - like the first perspective - as a black box. But here, the focus is on the determination of the input and output values. The structure orientated perspective helps to identify, understand and determine the internal structure of the system. This includes dynamic aspects like object, energy or information flows, processes or mechanisms of action. The hierarchical perspective considers the system from two sub-perspectives. The first one is a bottom-up perspective, which considers the system as part of another system. Through this, new comprehensive system delimitation that supports a holistic thinking becomes visible. The second perspective is a top-down one, which shows the system breakdown into its subsystems on different levels. In this scope a system of systems evolves, if systems are getting joined into one system, if a system gets integrated into another one, or if the system of interest was developed independent from the other parts and can realize its functions independently from a specific system context [23].

Like in fields of mechanical, mechatronic, electrical/electronic or software engineering, in systems engineering a lot of procedure models and methodologies have been developed over the last years as well [24]. *Haberfellner et al.* identified four essential basic ideas each procedure model should include [23]. These principles are:

- (1) starting from the rough and going to the details
- (2) consideration of alternative solutions
- (3) divide the process into chronological steps (phases)
- (4) use a formal guideline (problem-solving cycle)  
to find for each problem a solution

The first principle is related to several points already mentioned in the context of systems thinking. Thereby, the engineer should start with a large field of consideration for the system that will be restricted step by step. This includes the region of interest (the system and its environment) as well as the design of solutions. Starting with a system as a black box, the levels of detail and concretization will increase stepwise until all system entities and their connections are known (white box). With each level of solution concretization variability occurs. This means that there could be more than one solution to solve the problem. To obtain the best result, it is important to analyze, compare and evaluate these alternatives. In general, this could be alternatives on a very early level of the solution finding process, where each alternative is based on a different basic idea, or alternatives that are based on the same principle solution but disagree in the pre or final design. The third idea describes a macro strategy that extends the first two ideas. It divides the solution finding process into chronological steps and defines decision and correction nodes with the aim to reduce complexity as well as the risk of wrong decisions. This allows to jump back to a preceding phase and/or to focus on a different solution alternative. The problem-solving cycle describes a reusable micro-strategy, which can be used in each step of the development process. In general, it is based on

the identification of a problem, the search of alternative solutions strategies as well as their analysis, evaluation, and final selection. [23]

Systems engineering, in general, includes more than systems thinking and a procedure module which helps to turn a problem into a solution. Especially business needs, which are considered in the project management are as much as important as the technical needs. This paper, however, deals only with the aspects mentioned in this chapter.

## **4 Comparison Based on Essential Systems Engineering Aspects**

While the second chapter with the VDI 2206 guideline, the MVPE model and the mecPro<sup>2</sup> Architectural Framework gives a specific overview about design methodologies in the field of mechatronic and cybertronic systems, chapter three introduces essential aspects and principle ideas a system development process based on concepts of systems engineering should include. Table 1 gives an overview, whether and to what extent these fundamental aspect of Systems Engineering are included in the presented methodologies and approaches of chapter two.

As seen as in table 1, each approach includes one or more of the identified essential aspects and concepts of systems engineering. While the VDI 2206 is very abstract on the identified points, individual views of the mecPro<sup>2</sup> Architectural Framework can be assigned to the criteria. This is because the VDI 2206, on the one hand, looks at the entire development process from the requirements up to the finished product and, on the other hand, the VDI 2206 is a general guideline, which should guide the engineer during the development process. Whereas the mecPro<sup>2</sup> Architectural Framework is a specific methodological and model-based approach developed specifically for the system design phase of cybertronic systems. Although the MVPE model is an extension of the V-model from VDI 2206. It extends the scope of the view by the increase of the System Lifecycle Management Backbone to the entire life cycle and contains - with the SE-VPE method - a model-based procedure for the system design phase. While in the sense of cybertronic, mecPro<sup>2</sup> focuses primarily on a context-related description of the system, the focus of the SE-VPE method is mainly on the integration and administration of the essential system elements into a Product Lifecycle Management environment. This is especially evident in the rows ‘environmental orientated perspective’ and ‘impact orientated perspective’ of table 1. In general the SE-VPE method served as an important basis for the development of the mecPro<sup>2</sup> Architectural Framework.

## **5 Conclusion and Outlook**

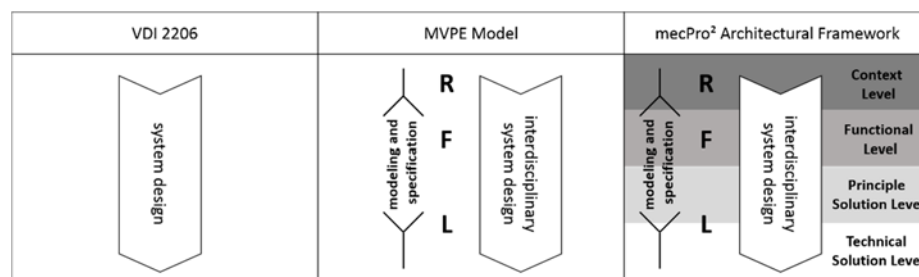
Based on the statement that systems engineering helps to reduce the complexity of today’s products [23] [24], this paper identified essential aspects and concepts a system engineering-based approach should include. Therefore, chapter four analyzed whether and to what extent these aspect are included in the selected methodologies for mechatronic and cybertronic development. Due to the fact that the introduced approaches build upon each other and represent enhancements, the level of fulfillment increases



**Table 1.** Comparison of design approaches based on essential systems engineering aspects

|  | <b>VDI Guideline 2206</b>   | <b>MVPE Model</b>  | <b>mecPro<sup>2</sup> Architectural / Model Framework</b>           |
|--|---|--|---|
| <b>Focus of the approach</b>   | Product Development Process   | Product Development Process and Product Lifecycle Management           | System Design   |
| <b>System Thinking</b>   |   |  |   |
| <b>Environmental oriented perspective</b>  |   |  |   |
| - Identification of the systems context of use   | Not explicitly described  | Implicit by the technical system requirements analysis (SE-VPE method) | Context Level (ContextDefinitionView)                               |
| - Identification of the system   |   |  | Context Level (ContextDefinitionView)                               |
| - Identification of the system boundary and its environment  |   |  | Context Level (ContextDefinitionView, ContextUseCaseView)           |
| <b>Impact oriented perspective</b>   |   |  |   |
| - Identification of the main system function   | System design (setting up the function structure)   | Functional flow definition (SE-VPE method)                             | Context Level (ContextUseCaseView)                                  |
| - Identification of the interactions between the system and its environment (function)                         | System design (setting up the functional structure)   |  | Context Level (ContextFlowView)                                     |
| - Identification of the interfaces between the system and its environment (form)                               | Not explicitly described  |  | Context Level (ContextInterfaceDefinitionView)                      |
| <b>Hierarchical oriented perspective</b>   |   |  |   |
| - Identification of the system entities  | System design (search for operating principles and solution elements)                         | Logical system modeling – black box view (SE-VPE method)               | Principle Solution Level<br>Technical Solution Level                |
| - Identification of the system entities' functions   | System design (setting up the function structure)   | Functional breakdown definition (SE-VPE method)                        | Functional Level (FunctionalBlockDefinitionView)                    |
| <b>Structure oriented perspective</b>  |   |  |   |
| - Identification of the interactions between system entities (function)  | System design (search for operating principles and solution elements)                         | Logical system modeling – white box view (SE-VPE method)               | Functional Level (FunctionalStructureView)                          |
| - Identification of the interfaces between system entities (form)  | Not explicitly described  |  | Functional Level (FunctionalStructureView)                          |
| - Creation of a system architecture (mapping of functional system structure to element based system structure) | Implicit during the search of operating principles and solution elements at the system design | F-L Allocation Modeling (SE-VPE method)                                | Functional Level<br>-<br>Technical Solution Level                   |
| <b>Basic Ideas of a Procedure Model</b>  |   |  |   |
| - Starting from the rough and going to the details   | Not explicitly described  | From black box to white box views along RFLP                           | Along the axes of concretization and detail                         |
| - Consideration of alternative solutions   | Included in the problem-solving cycle   | Not included   | Principle Solution Level<br>Technical Solution Level                |
| - Divide the process into chronological steps  | A guide for the basic procedure is offered by the V-model                                     | V-model and in system design along RFLP                                | Through the levels of the model framework and the process framework |
| - Use a formal guideline (problem-solving cycle) to find a solution for each problem                           | Reference to the SE problem-solving cycle of [8]  | Not explicitly explained by the SE-VPE method                          | Principle Solution Level<br>Technical Solution Level                |

with each approach. Nevertheless, each approach has its own right of consent. While the VDI 2206 describes a general guideline for the development of mechatronic systems, the MVPE model - especially with its SE-VPE method and the mecPro<sup>2</sup> Architecture Framework - represent methodical and model-based procedures for the design process. Since the mecPro<sup>2</sup> approach focusses exclusively on the design phase, there is no problem to integrate it into the interdisciplinary system design phase of the MVPE model (see figure 3). However, the SE-VPE method is not to be replaced by the mecPro<sup>2</sup> Architectural Framework, but it is an important alternative, especially for the specification of very complex systems.



**Fig. 3.** Integration of the mecPro<sup>2</sup> Architectural Framework into the system design phase

### Acknowledgments.

The authors would like to acknowledge the German Federal Ministry of Education and Research (BMBF) and the Project Management Agency Karlsruhe (PTKA) for founding the research project mecPro<sup>2</sup> (funding code 02PJ2573).

### References

1. Eigner, M.; Stelzer, R.: Product Lifecycle Management – Ein Leitfaden für Product Development und Life Cycle Management. 2. Ed., Springer, Berlin, Heidelberg (2009)
2. Broy, M.: Cyber-Physical Systems: Wissenschaftliche Herausforderungen bei der Entwicklung. In: Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme, pp. 17-32. Springer, Berlin (2010)
3. Lee, E.A.: Cyber Physical Systems: Design Challenges. University of California, Berkeley, Technical Report No. UCB/EECS-2008-8 (2008)
4. Eigner, M.; Muggeo, C.; Dickopf, T.; Faißt, K.G.: An Approach for a Model Based Development Process of Cybertronic Systems. In: Proc. Of the 58th Ilmenau Scientific Colloquium, Technische Universität Ilmenau (2014)
5. Cadet, M.; Meissner, H.; Hornberg, O.; Schulte, T.; Stephan, N.; Schindler, C.; Aurich, J.C.: Modellbasierter Entwicklungsprozess cybertronischer Produkte und Produktionssysteme - Grundlagen, erste Ansätze und weiteres Vorgehen. Stuttgarter Symposium für Produktentwicklung 2015, (2015)
6. Eigner, M.; Roubanov, D.; Zafirov, R.: Modellbasierte virtuelle Produktentwicklung. Springer, Berlin, Heidelberg (2014)
7. VDI 2206: Entwicklungsmethodik für mechatronische Systeme – Design methodology for mechatronic systems. Beuth, Berlin (2004)

8. Daenzer, W.F.; Huber, F.: *Systems Engineering – Methoden und Praxis*. Springer, Berlin (1994)
9. Brühl, A.P.: *Das V-Modell – Der Standard für die Softwareentwicklung*. Oldenbourg, München (1995)
10. Flath, M.; Kespohl, H.; Möhringer, S.; Oberschelp, O.: *Entwicklung mechatronischer Systeme*. In: *Entwicklungsumgebungen Mechatronik – Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme*, Bd. 80. HNI-Verlagsschriftenreihe, Paderborn (2000)
11. Eigner, M.; Gilz, T.; Zafirov, R.: *Proposal for Functional Product Description as Part of PLM Solution in Interdisciplinary Product Development*. In: *12th International Design Conference – Proceedings of the Design 2012*, pp. 1667--1676. University of Zagreb, Zagreb (2012)
12. Gilz, T.: *PLM-Integrated Interdisciplinary System Models in the Conceptual Design Phase Based on Model-Based System Engineering*. Band 13 Schriftenreihe VPE, Technische Universität Kaiserslautern, Kaiserslautern (2014)
13. Eigner, M.; Dickopf, T.; Apostolov, H.; Schäfer, P.; Faißt, K.G.; Keßler, A.: *System Lifecycle Management - Initial Approach for a Sustainable Product Development Process Based on Methods of Model Based Systems Engineering*. In: *11th IFIP WG 5.1 International Conference - PLM 2014*, pp. 287--300. Springer, Heidelberg (2014)
14. Sendler, U.: *Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM (Systems Lifecycle Management)*. In: *Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM*, pp. 1--20. Springer, Berlin, Heidelberg (2013)
15. Eigner, M.: *Modellbasierte Virtuelle Produktentwicklung auf einer Plattform für System Lifecycle Management*. In: *Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM*, pp. 91--110. Springer, Berlin, Heidelberg (2013)
16. Eigner, M.: *Das Industrial Internet*. In: *Industrie 4.0 grenzenlos*, pp.137—168. Springer, Berlin, Heidelberg (2016)
17. Eigner, M.; Dickopf, T.; Huwig, C.: *An Interdisciplinary Model-Based Design Approach for Developing Cybertronic Systems*. In: *14th International Design Conference – Proceedings of the Design 2016*, pp. 1647--1656. University of Zagreb, Zagreb (2016)
18. Eigner, M.; Dickopf, T.; Schulte, T.; Schneider, M.: *mecPro<sup>2</sup> - Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML*. In: *Tag des Systems Engineering 2015*, pp. 163--172. Hanser, München (2015)
19. Pohl, K.; Hönniger, H.; Achatz, R.; Broy, M.: *Model-Based Engineering of Embedded Systems – The SPES 2020 Methodology*. Springer, Heidelberg, New York (2012)
20. VDI 2221: *Methodik zum Entwickeln und Konstruieren technischer System und Produkte – systematic approach to the development and design of technical systems and products*. Beuth, Berlin (1993)
21. Ponn, J.; Lindemann, U.: *Konzeptentwicklung und Gestaltung technischer Produkte – Systematisch von Anforderungen zu Konzepten und Gestaltungsformen*. Springer, Berlin, Heidelberg (2011)
22. Crawley, E.; Cameron, B.; Selva, D.: *System Architecture – Strategy and Product Development for Complex Systems*. Pearson Higher Education, Hoboken, New Jersey (2016)
23. Haberfellner, R.; de Weck, O.; Fricke, E.; Vössner, S.: *Systems Engineering – Grundlagen und Anwendung*. Orell Füssli, Zürich (2012)
24. INCOSE: *Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons, Hoboken, New Jersey (2015)
25. Estefan, J.A.: *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. Pasadena, CA, USA. International Council on Systems Engineering (INCOSE) (2008)