

E/E-Product Data Management in Consideration of Model-Based Systems Engineering

Marco W. GROLL and Dominik HEBER¹

Daimler AG, Germany and University of Twente, Netherlands

Abstract. This paper presents objectives for permeable electric/electronics product data management for mechatronic products in consideration of model-based systems engineering from the early product development phase till a lifecycle management. Idiosyncrasies of mechatronic products, requirements engineering, model-based systems engineering, artifact-orientation, and interconnections of artifacts are evaluated and postulate objectives, how artifacts have to be designed in order to support the linkage of model-based systems engineering and product data management (PDM). The objectives, derived from the different theories and requirements to foster permeable PDM, are: i) Identify all existing norms for the development of mechanical, electronic, and software aspects and elaborate how information artifacts have to be defined. ii) (Textual) Requirements have to be technically feasible to be linked to information artifacts and system models already in the early development phase. iii) System models have to be aligned to information artifacts from the models' creation onwards and standardization in exchange formats has to be ensured. iv) Information artifacts with own lifecycles shall alleviate PDM in the early product development phase. v) Interconnections shall ameliorate associativity through capturing process information between single artifacts. A first concept is presented, visualizing the aforementioned objectives and their contribution in the early development process of mechatronic products, how a permeable PDM might be achieved.

Keywords. Electric/Electronics, Model-Based Systems Engineering, Interconnections, Permeable Product Data Management, Mechatronics, Requirements Engineering

Introduction and Motivation

Due to a steady increase of variants and a higher degree of digitalization within automobiles, demands towards an IT landscape within the early phase of the product development process rise tremendously. Within roughly one decade, the number of control units in a premium middle-class car doubled [1]. This digitalization of the automobile industry partially stems from the trend towards connectivity: on the one hand the user wants to be connected with his car via smartphone; on the other hand cars shall be connected amongst each other, the so-called car2x. Additionally, more and more assistance systems, which make use of mechatronic systems, find entry to the automobile realm [2]. Mechatronic systems are the composition of mechanical, electronic and software components [3]. Within mechatronic systems, the proportional share of software advances continuously and hence yield to an augmentation of functional complexity [3]. This is demonstrated by the figure that 50 to 80% of

¹ Corresponding author, E-Mail: Dominik.Heber@Daimler.com

innovations in the automobile industry originate from software [3]. Moreover, an ever more globalized collaboration in the development of automobiles requires interconnected workflows and employment systems [3].

The described increase of complexity in the product development process infers needs for action particularly for product data management (PDM). PDM describes the design engineering phase, including product planning and development, and is inherent to the product lifecycle management (PLM) [3]. Due to this complexity, in today's automobile industry requirements engineering (RE), model-based systems engineering (MBSE) and PDM often are separated organizationally and each domain also has its own IT-system. The impermeability between document-based RE, MBSE and PDM in today's concurrent engineering IT-landscape still constitutes a major barrier in a continuous workflow and leads to increased time and effort as well as redundancies and error-proneness [3].

This present paper aims to enrich the concept of a permeable workflow in the early phase of product development and how a consequent PDM of information artifacts supports alleviating the increased complexity in the automobile engineering, particular in the mechatronics.

Section one will deal with idiosyncrasies of mechatronic products and methods which are used in the development process. Then, based upon the idiosyncrasies and current methods in the product development process, objectives for a permeable PDM will be stated. Section two presents a first concept of artifact-oriented interconnection product documentation for mechatronic products in consideration of model-based systems engineering. Section three closes with planned further proceeding and an outlook.

1. Implications for a Permeable Product Documentation Management

1.1. Idiosyncrasies of Mechatronic Products with Regard to Product Development Process

In the last decade the amount of electric and electronic parts, control units and systems within an automobile surged [1]. As stated above, mechatronic systems today consist of electronic, mechanic and software components and underwent a transition from purely mechanic and electronic systems towards a composition also with software parts, as shown in Figure 1. To electric/electronic (E/E) parts software is not necessarily immanent, e.g. sensors and actors. Yet, control units, which coordinate amongst others sensors and actors, in a vehicle increased their importance and quantity due to an augmented relevance of connectivity and assistance systems in an automobile (cf. section 1). Therefore, this elaboration will focus on so-called "intelligent" E/E components, i.e. electronic control units (ECU) or mechatronic components.

Hence, the need for methodical process models in the product development of mechatronic products is more crucial than ever. For that reason, there is a blossoming of different methods and process models intending to handle the extreme complexity of developing mechatronic products. The most popular process model in the automobile industry with regard to the development of mechatronic products is the V-Model according to the design methodology for mechatronic systems (VDI 2206) [4].

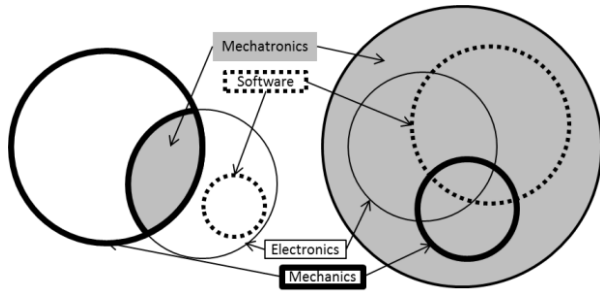


Figure 1. Transition of manufacturing engineering from yesterday (left) to tomorrow (right) (according to [3]).

Bender [5] extended the V-Model to a three-layered V-Model, addressing the different domains within mechatronic products separately. In the two upper levels, i.e. vehicle and system level, development occurs jointly. Successively, sub-systems and components are developed domain-specifically (cf. Figure 2). Precisely this separation of development in domain-specific tracks, which are often divided technically and organizationally, is what makes the development of mechatronics so challenging.

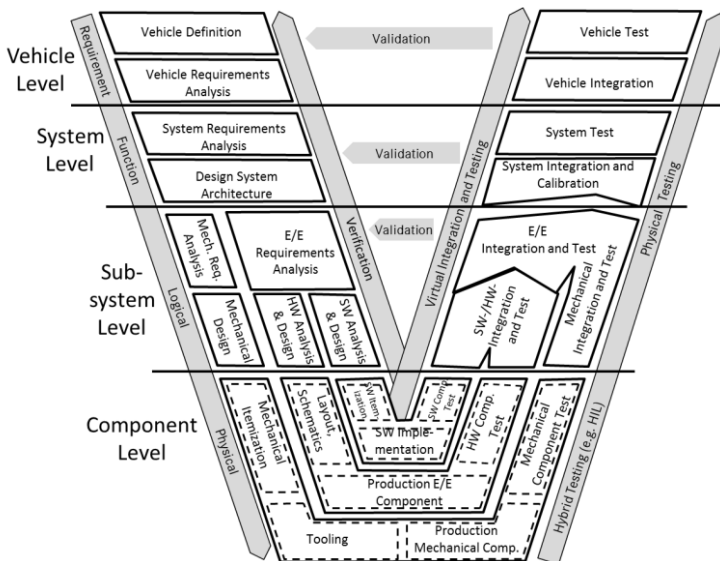


Figure 2. V-Model of mechatronic product development extended for systems development and applied to automobile development (according to [3], [5], [6]).

With regard to a permeable PDM, the idiosyncrasies of mechatronic products postulate objective i):

Capture all mechanical, electronic and software aspects which are necessary throughout the product development process to foster a permeable, model-based PDM. Identify all existing norms and elaborate how information artifacts have to be defined.

1.2. Requirements Engineering

Requirements engineering is defined by the thorough documentation of requirements of behavior, quality, and integration in a technical manner so as to assure the quality of the product and satisfy the customer's expectations prematurely. Requirements engineering is an incremental, iterative, and cooperative process targeting at the determination, analysis, understanding, and definition of requirements. Moreover, requirements management declares precise development aims to recognize and track errors and changes earlier in order to develop more efficiently. Requirements management is intrinsic to requirements engineering [3].

For the purpose of a precise description of requirements and their management during the development process, conjunct attributes are necessary. According to Eigner et al. [3], requirement attributes are determined by a name, associated semantics, and a range of values. Those attributes are assigned to categories [3]: identifiability, relations of context, aspects of documentation, aspects of content, aspects of congruence, aspects of validation, and aspects of management. Each attribute can be assigned a scheme of attribution. Schemes of attribution describe predefined and comprehensive values, such as "new", "in process", or "implemented" in case of requirements management [3].

With regard to a permeable PDM, the phase of requirements engineering postulates objective ii):

Requirements have to be designed technically and with regard to contents so that already in the early product development phase (mostly text-based) requirements can be associated to information artifacts and system models which can be used throughout the entire product lifecycle.

1.3. Model-Based Systems Engineering

A decisive impediment in the product development method of Bender [5] is that a model-based product development is not explicitly addressed. The common product development is an integrated, multi-disciplinary activity comprising all steps the product undergoes in its development (such as production, operation, and disposal) and its lifecycle to the point of supply chain [3]. Today the means of digitalization of the product development process and visualization of the results of each step of the development process is crucial for product quality and the entire product lifecycle management [7]. Therefore, the virtual product development gains more importance. For this purpose, IT-solutions are utilized in order to support and optimize the creation and documentation of work results and to ascertain that those results are available throughout the entire product lifecycle. If the description within each particular phase of the lifecycle is based upon a formal language (a formal language is an abstract language which focuses on mathematical or physical deployment [3]), digital models are built and placed at the disposal by transformation for the next phases, then the IT-process chain is denominated as model-based [3]. Hence, the model-based virtual product development (MVPD) aims at the reduction of physical prototypes by deployment of permeable, computer-supported, formal modelling and documentation. Moreover, enabling reutilization and transmission of models across all phases of the product lifecycle also is crucial to MVPD [3].

Systems engineering (SE) is defined as an inter-disciplinary, document-based approach for the development and implementation of technical and complex systems in

order to assure a high-value fulfilment of stakeholders' and customers' requirements throughout the entire lifecycle [3]. Hence, SE does not solely focus on the development of a single mechatronic product, as in contrast to Bender's V-Model [5], but more on the development of complex systems which include e.g. several mechatronic components. Model-based systems engineering (MBSE), as well as MVPD, makes use of a formal language and designated system models to determine the product within the product development process in each of its phases and also in downstream processes [3]. Figure 2 already extends Bender's [5] V-Model to that effect that the different development phases for model-based systems engineering are considered: requirement, function, logical, and physical. Additionally, hybrid testing, e.g. hardware in the loop (HIL), and physical testing of model-based systems are displayed in Figure 2. However, testing is not in scope of this elaboration.

Eigner et al. ([8], [9], and [10]) present an approach how model-based systems of different disciplines can be managed throughout the entire lifecycle so that all of the time a virtual image, i.e. a so-called *digital twin*, is available. This approach is called systems lifecycle management (SysLM). SysLM depicts the information management in general which extends common product lifecycle management (PLM) by an explicit consideration of upstream and downstream phases of development [8]. Therefore, all disciplines are in scope. The PLM backbone shall incorporate the entire system lifecycle management [8].

Gilz [11] describes a method for integration of discipline-specific MBSE data in a PLM backbone by usage of a functional product description (FDP). Functions are common information artifacts in an interdisciplinary context. The main artifacts of the FDP are: system requirements, system functions, logical system elements, and physical system elements [11]. Gilz [11] further highlights in his outlook the need for a combination of system models and configuration as well as variant models.

With regard to a permeable PDM, the concept of model-based systems engineering postulates objective iii):

System models have to be aligned to information artifacts from the models' creation onwards throughout their entire lifecycle (systems lifecycle management). Permeability between IT-systems and development domains has to be ensured.

1.4. Artifact-Oriented and Product Data Management

In software development, the need for efficiency enhancements with regard to reutilization of previously developed so-called components is nothing new [12]. By means of those components, small- or large-scale software can be composed by reusing the previously developed software objects. Due to object and component have different meanings concerning which domain is in scope, the terminology *artifact* will further on depict a piece of information with a unique identifier within IT-systems in the virtual product development process. For the distinction of objects and components in software development see Brown [12]. Moreover, a component in automobile industry commonly refers to a mechanical element or part which fulfills a dedicated purpose in an automobile. However, even within one company there exist a plethora of different meanings of a component [13]. Hence, the terminology artifact is chosen in order to distinguish from sole software development and from the term mechanical component which usually is used in automobile industry. Product team data management (TDM)

and PDM software implements an innovative technique using single software artifacts which can be managed independently throughout the entire lifecycle of ship development and production [14]. Each artifact has its own lifecycle with its revisions and status which is not limited by an assembly item's lifecycle. Hence, isolated revisions and change management of each artifact within a product is feasible, apart from the assembly [14]. Additionally, [14] depicts that so-called collaborative designs, e.g. a certain ship model, can be defined company-wide. Partitions, intrinsic to the collaborative design, represent a structural breakdown of the product and can be formed manually or via recipe rules. Partitions are formed by different artifacts, which are linked to the corresponding partition, and display the traditional hierarchical view of an engineering bill of material (E-BOM) [14]. Partitions are separated into functional (e.g. different (mechanical) systems for different purposes), spatial (e.g. different zones of the final product), and physical (e.g. part x). Data representation in the artifact-oriented PDM is semi-non-hierarchical, i.e. artifacts are stored in a vault and managed separately (see above). However, for purposes of usability, collaborative designs are defined but artifacts have no specific structural position; hence, the non-existence of hierarchy is partially diluted, i.e. "semi-non-hierarchical". Conversely, traditional PDM systems have a single, semi-flexible hierarchical product structure due to a configurable structure level-by-level. This single, semi-flexible hierarchical product structure is comparable to a book library, because books are retrieved from a shelf and returned to the same shelf, as well as products are loaded out of a BOM structure and returned at the exact same position during development. Artifact-oriented PDM incorporates recipe rules for product variant structuring which is similar to traditional, hierarchical PDM systems. Both use Boolean algebra to create code conditions in order to in- or exclude further configuration options, such as partitions, modules, etc. [14].

With regard to a permeable PDM, the concept of artifact-oriented PDM postulates objective iv):

Information artifacts shall alleviate handling complexity in the early product development process by owning their own attributes and lifecycle. Specific designs, e.g. collaborative, functional, spatial, and physical, are generated by combination of single artifacts. Artifacts shall be managed in a common PLM backbone.

1.5. Interconnections

The concept of artifact-oriented PDM and traditional, i.e. hierarchical, PDM use recipe rules, based upon Boolean algebra, for product and variant configuration. In hierarchical product structures products still are depicted as a sum of single parts. Furthermore, structural associativity between parts and assemblies are, if at all, only limited represented in traditional PDM. The Boolean algebra codes can mutually exclude components within an assembly. However, management information between components on the same structural level are not provided [15]. This deficit of associativity between components on the same level, in case of traditional PDM, stems from the fact that structural togetherness mainly is described through parent-child relationships between assemblies and their intrinsic components in the form of "is part of" or "belongs to". Interconnections mitigate this issue by describing a product as parts and interconnections in a web to enable an integrated product model (cf. Figure 3) [15]. This allows for an overview which captures all elements with their structural

associativity, comparable to a systems view (cf. section 1.3.). In comparison to the artifact-oriented PDM method with a semi-non-hierarchical product structure, interconnections yield no further advantage with respect to product decomposition and structure. Yet, interconnections also comprise process information, such as tooling, assembly order or factory planning. For this reason, the concept of interconnections can describe more lifecycle phases than the artifact-oriented method which only limitedly includes tooling and factory planning. A drawback of the interconnections documentation is that it cannot consider computer-aided design (CAD) because it only describes the lowermost component decomposed in its parts and interconnections.

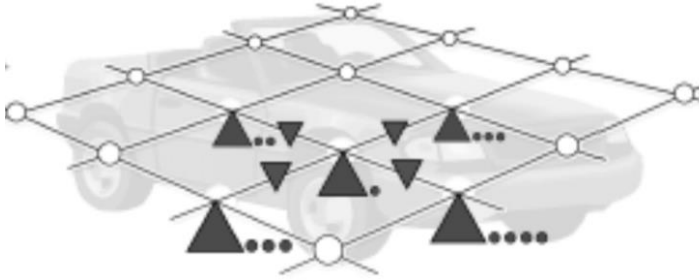


Figure 3. Interconnection product structure decomposition [15].

With regard to a permeable PDM, the concept of interconnections postulates objective v):

Ameliorate information for associativity through process information captured in interconnections between single artifacts. Interconnections shall mitigate disruptions between requirements engineering, model-based systems engineering and product data management by providing additional insights in systems and their associativity.

2. Approach to the creation of a concept

Information artifacts, already generated in the requirements engineering (RE) phase, will be used in this elaboration to address the issue of continuous and permeable product documentation (cf. section 1.2 and 1.4). During RE phase, artifacts are associated (*scope*, displayed by the crosshairs in Figure 4) with the first drafts of vehicle outlines, e.g. architecture (see Figure 4). Besides, access regularities for development engineers are defined (displayed by the lock in Figure 4).

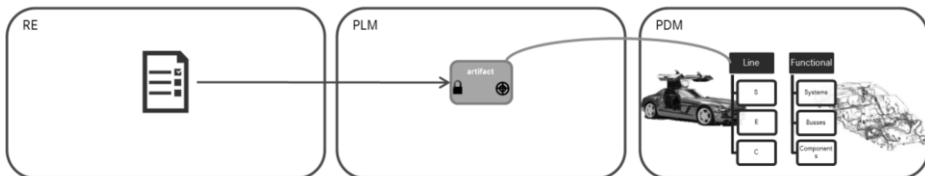


Figure 4. Associativity of information artifacts in requirements phase.

Those information artifacts each are deposited independently in a multidisciplinary repository, containing different, specific attributes regarding their lifecycle, access authorization, geometric position, and configuration range (cf. section 1.4). Model-based systems engineering engages those previously defined artifacts, which have been enriched with further information, and simulates the interaction of the components in their specific system and for their specific purpose, e.g. within a computer-aided software engineering model, simulation model, manufacturing computer-aided design (CAD) model, electronic CAD model, and the overall system model, which depicts the interdependency between the single models and their global functionality, as depicted in Figure 5 (cf. section 1.3).

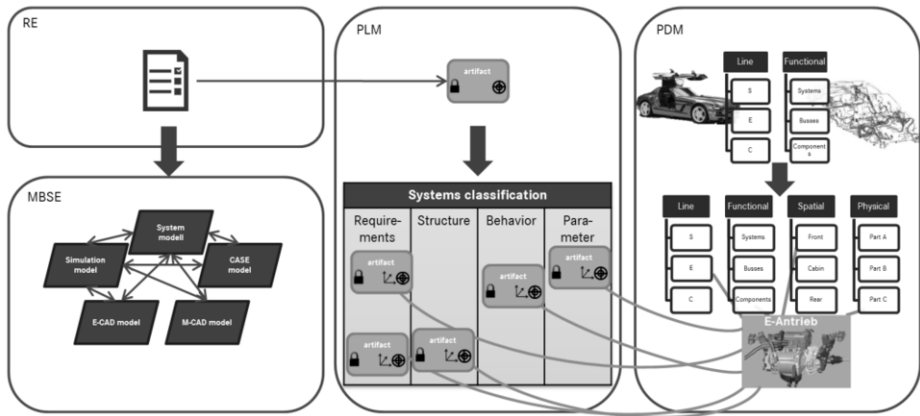


Figure 5. Associativity of information artifacts in model-based systems engineering phase.

Via interconnections, immanent linkages that incorporate further qualitative and quantitative product- and process-related information, simulation models, structural breakdowns and views, as well as information artifacts are associated to each other (cf. section 1.5). Hence, a reutilization in following simulations can be indemnified without re-specification. In the PDM, a semi-hierarchical product structure and an ad-hoc, user-specific representation given the individual purpose, mitigate documentation complexity and effort (cf. section 1.4). Therefore, information artifacts are associated through interconnections to multiple physical components, as well as directly to functional systems, spatial divisions of vehicles or entire model lines (cf. section 1.5). Each engineer administers the information artifacts in the multidisciplinary repository, which therefore functions as a single source where also variant and configuration management takes place (cf. section 1.3 and 1.4). Multidisciplinary repository, single source, and PLM backbone are used as synonyms. Whenever a new vehicle is created, the already pre-defined information artifacts in the PLM backbone can be deployed to be re-associated to this new vehicle without entire re-development, as depicted in Figure 6.

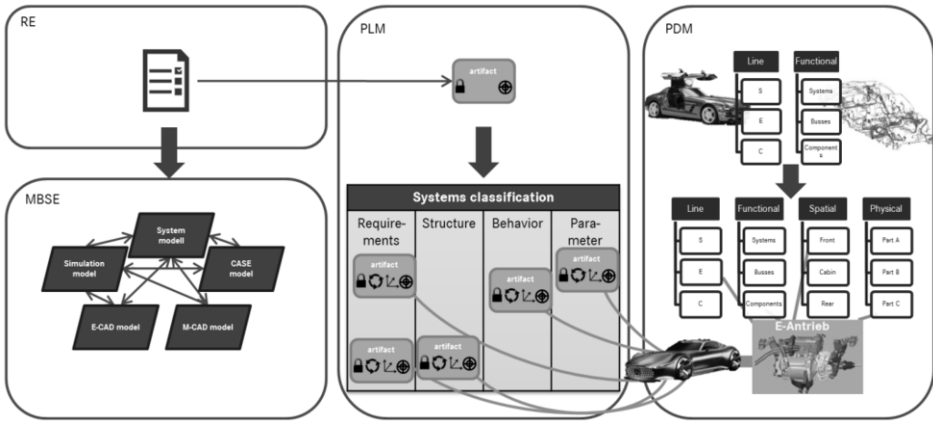


Figure 6. Re-utilization of pre-defined artifacts for a new vehicle.

The target situation is shown in Figure 7. Objective i) would be addressed inherently by the design of information artifacts. Objective ii) is captured in the RE phase whereas objective iii) is also already pre-defined in the RE phase and further specified during the dedicated MBSE phase. Objective iv) has to be taken into consideration throughout the entire conceptual work, particularly with regard to the PLM backbone and requirements stemming from discipline specific designs and partitions in the PDM system. Objective v) is crucial for the linkage of MBSE, PLM, and PDM. Moreover, objective v) supports an improved allocation of artifacts in their specific destination, e.g. interconnected with a system, a spatial area, or a physical component.

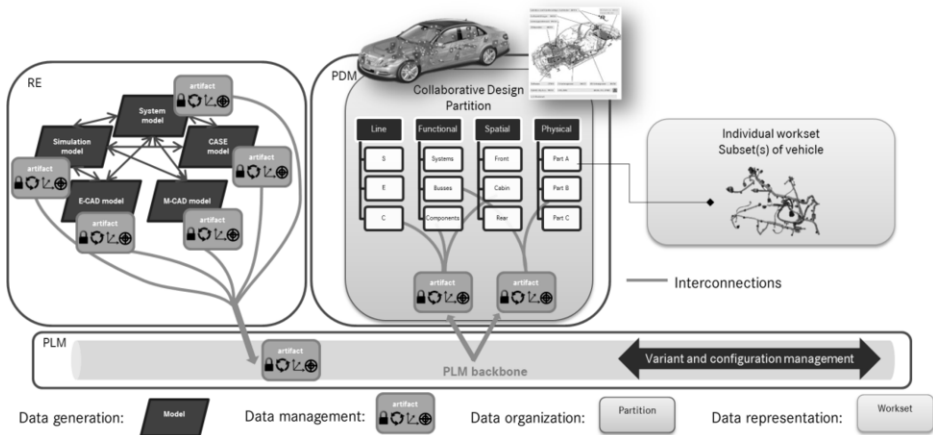


Figure 7. Conceptual target situation of artifact-oriented interconnection product documentation in consideration of model-based systems engineering.

3. Outlook

Due to being a conceptual paper, there is still a lot to consider. However, the main theories have been highlighted and also which objectives stem from them with regard to artifact-oriented interconnection product documentation for mechatronic products in consideration of model-based systems engineering. Additionally, state of the art norms, methods, and concepts related to each presented phase and method of product development have to be observed. According to [16], further literature research has to be conducted to additionally clarify the research questions. Then, an empirical data analysis follows, e.g. a stakeholder rally. This is the descriptive study 1 and it fosters the understanding by building a reference model and defines success criteria. Afterwards, assumptions, experiences, and synthesis form the prescriptive study which results in an impact model and support evaluation. The descriptive study 2 uses real data in order to evaluate the impact model's success and predict further implications.

References

- [1] Daimler AG, *Internal Documentation*, n.a.
- [2] W. Siebenpfeiffer, *Vernetztes Automobil*, Springer Vieweg, Wiesbaden, 2014.
- [3] M. Eigner, D. Roubanov, R. Zafirov, *Modellbasierte virtuelle Produktentwicklung*, Springer Vieweg, Berlin, 2014.
- [4] Verein Deutscher Ingenieure, *VDI-Richtlinie 2206: Design methodology for mechatronic products*. Beuth Verlag, Berlin, 2004.
- [5] K. Bender, *Embedded Systems – qualitätsorientierte Entwicklung*, Springer, Heidelberg, 2005.
- [6] ProSTEP iViP, *Smart Systems Engineering*, <http://www.prostep.org/en/projects/smart-systems-engineering.html>, last accessed: April 7, 2016.
- [7] A. Katzenbach, S. Handschuh, R. Dotzauer, A. Fröhlich, Product Lifecycle Visualization, in J. Stjepandic, W. J. C. Verhagen, N. Wognum, *Concurrent Engineering in the 21st Century*, Springer International Publishing, Switzerland, 2015.
- [8] M. Eigner, C. Muggeo, H. Apostolov, P. Schäfer, Kern des System Lifecycle Management - im Kontext von Industrial Internet mit Industrie 4.0 und Internet der Dinge und Dienste, *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 111 (2016), 63-68.
- [9] M. Eigner, K.-G. Faißt, H. Apostolov, P. Schäfer, Kurzer Begriff und Nutzen des System Lifecycle Management - im Kontext von Industrial Internet mit Industrie 4.0 und Internet der Dinge und Dienste, *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 110 (2015), 475-478.
- [10] M. Eigner, H. Apostolov, T. Dickopf, P. Schäfer, K.-G. Faißt, System Lifecycle Management - am Beispiel einer nachhaltigen Produktentwicklung nach Methoden des Model-Based Systems Engineering, *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 109 (2014), 853-860.
- [11] T. Gilz, *PLM-Integrated Interdisciplinary System Models in the Conceptual Design Phase Based on Model-Based Systems Engineering*, Dissertation, Technical University of Kaiserslautern, Kaiserslautern, 2014.
- [12] A. W. Brown, *Large-Scale, Component-Based Development*, Prentice Hall PTR, Upper Saddle River, 2000.
- [13] Daimler AG, *Internal Documentation*, n.a.
- [14] Siemens PLM Software, *Handbook Teamcenter 10.1*, Siemens Industry Software, Plano, 2014.
- [15] M. W. Groll, *Interconnected Based Product and Process Documentation*, Dissertation, University of Twente, Enschede, 2008.
- [16] L. T. M. Blessing, A. Chakrabarti, *DRM, a Design Research Methodology*, Springer-Verlag, London, 2009.