Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 84 (2019) 451–456

www.elsevier.com/locate/procedia

29th CIRP Design 2019 (CIRP Design 2019)

Enabling complexity management through merging business process modeling with MBSE

Christian Konrad*, Georg Jacobs, Rik Rasor, Ricarda Riedel, Tim Katzwinkel, Justus Siebrecht

iMSE RWTH Aachen University, Steinbachstraße 54B, 52074 Aachen, Germany

* Corresponding author. Tel.: +49 241 80-27355; fax: +49 241 80- 22286. *E-mail address:* christian.konrad@imse.rwth-aachen.de

Abstract

Integral products are characterized by multiple functional use of components, resulting in more compact and lighter products. However, those benefits come along with a significant increase in development process complexity. This complexity represents a significant risk in the value-creation chain. For the efficient fulfilment of customer needs and market requirements, strategies are required to cope with the complexity of integral products. A promising approach in this context is Model Based Systems Engineering (MBSE). Through the formalized application of modeling, MBSE allows the development of a system model to support the development process. The model enables a visualization of the process complexity and thus an identification of the essential complexity drivers. This knowledge is mandatory for the coordinated initiation of countermeasures. However, MBSE is not yet fully integrated in today's product development processes (PDP). One reason for this is an insufficient interconnection between system models and business processes. This paper presents an approach for merging MBSE with business processes. The procedure comprises the modeling of exemplary business processes in a product data management system (PDMS). These processes are linked to a MBSE modeling tool via a RESTful API. Synergetic potentials of this connection are pointed out, based on scenarios from requirements management, product configuration and risk management. Finally, a discussion takes place regarding the degree of maturity to master complexity in business processes with MBSE.

Keywords: Model Based Systems Engineering; Product Development Process; Complexity Management; Integral Product Architecture

1. Introduction

Shorter product life cycles, changing requirements, increasing complexity and increasing individualization of products are today's challenges for mechanical engineering companies [1,2,3]. The demand for cost effective product designs and high-performance requirements (e.g. low weight, small build space, sustainability) make integral product architectures an attractive concept for design engineers. While the overall performance of integral product design can be higher than comparable modular product architectures, the system and process complexity of integral designs is more difficult to control. This area of tension can result in an increasing number of variants and a multitude of order-specific elements. Customer individuality generates high coordination efforts and engineering change costs in the value-added process. This applies especially to products with integral architectures [4,5]. The increasing complexity in industrial development requires new strategies to control complexity and increase efficiency in development processes of technical products [6], mass customization manufacturing systems [7] and product service systems [8].

In the context of product development, Model Based Systems Engineering (MBSE) is a promising approach for managing the complexity of technical systems. Accompanying the product development process, a system model is set up as an information backbone. Requirements, application scenarios, and product structure information are modeled in specialized software tools using interactive diagrams for visualization.

10.1016/j.procir.2019.04.267

In addition to the complexity of the technical system, the complexity of the organizational system must also be manageable. A common approach for managing the complexity of business processes is business process modeling (BPM). Similar to the MBSE approaches, BPM uses standardized, graphical modeling languages like Business Process Model and Notation (BPMN) for the modeling of processes and workflows. The modeled processes define the stakeholder's activities within the development process while considering required input data and generated result data as well. In large scale companies, modeled processes are usually implemented in software tools like Product Data Management Systems (PDMS) to ensure process security and controllability and to support the stakeholder in the execution of his activities.

In the corporate context, the complexity of the technical system is coupled to the organizational system [4]. In order to make the resulting overall complexity manageable, the individual approaches to the control of technical or organizational system have to be merged in a practical way. So far, there are hardly any approaches for a practicable linking of MBSE with BPM. In the context of this paper, first of all a methodical procedure for the practical merging of MBSE with BPM will be presented. Furthermore, this methodical procedure is applied to a use case in a typical large scale company IT-infrastructure for a feasibility study.

2. State-of-the-Art and Need for Action

The aim of MBSE and BPM is to improve complexity management, communication, specification and development process controllability in their areas of application. [9,10]

MBSE is the formalized application of modeling to support requirements capture, development, verification and validation of a system over the entire product lifecycle [1,2]. The product development is supported by a central system model [3]. In the sense of system-theoretical abstraction, a technical system model is an abstract representation of a product.

System models can be built with the help of modeling languages. These languages usually consist of a specific notation syntax, an abstract content syntax and semantics. The notation syntax defines the appearance of the symbols, characters and expressions used. The abstract content syntax is a set of rules and principles that regulates the availability of notation elements and the arrangement rules based on them. Semantics describes the meaning of the available notations that the modeling language should express. [11,12]

The system model should contain all relevant data about the product as well as the product context, so that it forms the central information backbone for all stakeholders of the development process [13]. The current focus of MBSE research lies in the areas of requirements management [2,14,15], system simulation and validation [16,17,18,19] as well as system development [20]. Exemplary, methodological approaches for building MBSE system models are FAS [21] and SYSMOD [22]. The utilized system models should be managed, linked and developed over the entire product lifecycle [2,23]. Although the mentioned methods consider structural relations, they do neither explicitly address the physical parameter level of the system nor the integration of MBSE into BPM.

Business process management comprises the identification, planning, design, implementation, documentation, control and improvement of business processes [10]. The processes are continuously coordinated with the corporate strategy in order to achieve the business objectives [10]. Various BPM approaches exist [9,10,24]. Similar to the MBSE approach, specialized modeling languages are used to describe complex processes with visual representations. One modeling language in that context is BPMN. Use cases of BPMN are the simulation, analysis and implementation of business processes [25]. The superior goal of BPMN is to support the basic understanding of company structures and business processes by offering a common standardized visual language. Like MBSE, the BPMN standard is deliberately defined independent of methods or modeling concepts [26]. An implementation of MBSE models in the management of today's business processes is not yet state-of-the-art [6].

While the MBSE approach focuses on the product and its domain-specific properties, business processes primarily map processes in the corporate context. In order to make the complexity of product development processes (PDP) manageable in an enterprise environment, a methodical merge of business processes with the technical system is necessary.

3. Method

To merge the technical system modeled with MBSE and the business process modeled with BPM, an approach is required that links both modeling languages. BPM and MBSE can each be understood as domain-specific views in the overall PDP. In the context considered in this paper, the relationships of the product and the process model in particular have to be modeled. Therefore, three classes of relationships can be differentiated. These relationships are shown in Fig. 1.

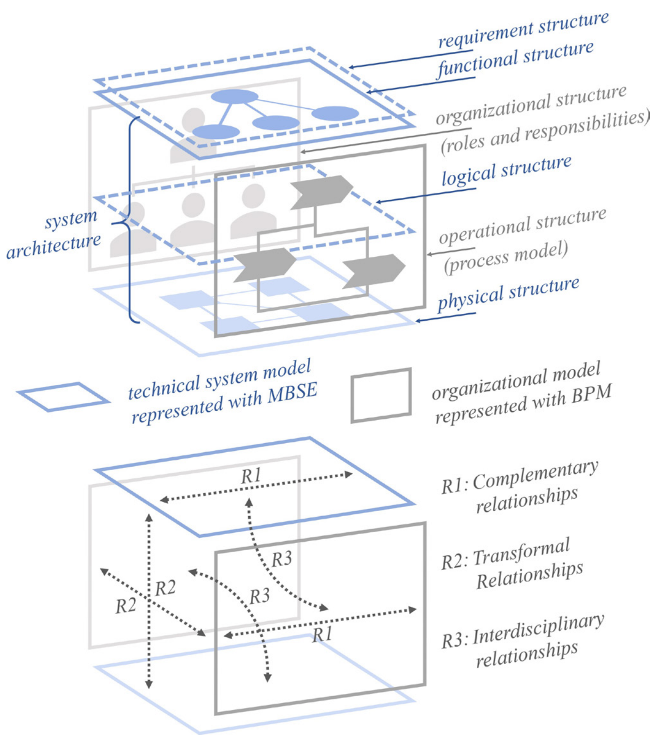

Fig. 1. Principle relationships between the layers of MBSE and BPM.

The BPM views consist of the two layers “organizational structure” and “operational structure”. The MBSE views consist of the four layers “requirement”, “functional structure”, “logical structure” and “physical structure”. Within Fig. 1 the possible relationships are classified:

- Complementary relationships (R1): layer-internal relationships of the elements within one domain-specific layer, e.g. interdependencies between physical components.
- Transformational relationships (R2): cross-layer relationships between two or more model layers of one specific domain, e.g. assignment of a role to a process.
- Interdisciplinary relationships (R3): cross-layer and cross-domain relationships between elements of two or more layers of different domains, e.g. assignment of an engineering change process to affected components.

The functional scope of the existing domain-specific modeling tools provides no standardized interface for the practical mapping of the relations between the domain-specific views. The method presented in this paper follows the approach of merging the MBSE and BPM views in a PDMS as superior platform.

According to the Product Lifecycle Management (PLM) paradigm, corporate strategies are often based on three pillars: processes, stakeholders and data. Usually, the implementation of such corporate strategies takes place in so-called PDMS. State-of-the-art PDMS have proven their worth for the management of product and process views over the last decade [27]. PDMS are currently used for data management and process integration between the specific applications of the PDP, e.g. mechanical CAD, simulation data or requirements management. Nevertheless, data management usually does not include dynamic access to simulation models or MBSE models and their parameters. Therefore, a framework is presented which enables the merging of MBSE and BPM approaches in a PDMS. The framework consists of four steps:

- Analysis of domain-specific modeling languages and depth of the merge
- Analysis and setup of the company-specific information technology (IT) infrastructure
- Linking of IT systems for assignment of PDMS functions for merging MBSE and BPM
- Determination and implementation of interdisciplinary relationships between MBSE and BPM

Following this framework, the feasibility of the design of interdisciplinary relationships and the resulting potentials will be evaluated in chapter 4 based on a use case.

3.1. Analysis of domain-specific modeling languages and depth of the merge

To merge BPM with MBSE in a superior platform, first the corresponding elements of the modeling languages and their relationships must be identified and analyzed. For example, those elements can be system model parameters, workflows and their interdependencies. Fig. 2 shows an overview of the domain-specific model elements of the technical system, MBSE and BPM on equivalent observation level.

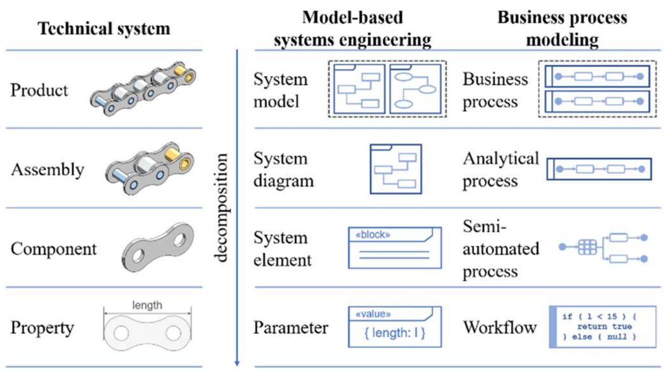

Fig. 2. Overview of the domain-specific elements on equal observation level.

The product, the system model in MBSE and the business process in BPM are on equal observation level and can each be further decomposed hierarchically. The system model represents the technical view of the product and consists of the system model layers, see Fig. 1. The system model layers and their interconnections are described with the help of system diagrams. System diagrams are composed of system elements and their interactions. System elements are characterized by parameters.

The business process is the organizational and operative view on the product and is hierarchically composed of analytical processes, semi-automated sub-processes and workflows.

The presented analysis is an important preceding step for the design of interdisciplinary relationships in chapter 3.4.

3.2. Analysis and setup of the company-specific IT infrastructure

The practical added value of a formal link between BPM and MBSE based on hierarchical equivalent models is tied to the feasibility of linking on the tool level. Here, the challenges of the typical best of breed strategy are quickly revealed when selecting IT tools. The high flexibility of large scale IT solutions is at the expense of resources, charges for the integration of the different providers as well as the management of interfaces and administration expenses [28]. In order to check the feasibility and potential of linking BPM and MBSE, the specific company IT and tool structure must be analyzed and integrated into the procedure.

As part of the preparatory work for this publication, suitable exemplary tools were selected for the practical implementation of MBSE and BPM in a state-of-the-art business context. The Software Cameo Systems Modeler (CSM) by No Magic, Inc. was selected as the MBSE SysML implementation tool. For process and stakeholder models the community edition of the PDMS Aras Innovator by Aras Corporation was selected. Both software tools had no suitable software interfaces at the time of consideration. However, the tools were able to adapt application programming interfaces (APIs) with extensions and customizations.

3.3. Linking of IT systems for assignment of PDMS functions to merge MBSE and BPM

The linking of IT systems requires a clear strategy for structuring information and the associated data models. Therefore, the following questions have to be answered:

- Which elements of the existing modeling languages can be integrated into the PDM platform?
- How can the structure of domain-specific layers be represented during the integration into the superior platform?
- Which form of visualization is suitable for relevant information in the PDMS?
- How can the programming interfaces between the tools be designed?

The analysis of the model elements and structures is strongly dependent on the use case as well as on the selected tools. In contrast, the general procedure for designing the programming interface is suitable for all tools and crucial for successful linking. Therefore, the procedure is described in more detail. To synchronize the information objects, a programming interface between the PDMS and the domain-specific modeling tools is required.

Comparable to the use of standardized modeling languages, vendor-neutral interfaces should be used. In this context, the programming paradigm Representational State Transfer (REST) has become widely accepted in recent years. REST offers a well-established approach to how distributed systems can communicate with each other. A corresponding RESTful programming interface implements REST-compliant standardized procedures such as Hypertext Transfer Protocol (HTTP), Uniform Resource Identifier (URI) and JavaScript Object Notation (JSON) [29]. The tools used in this work have REST interfaces and the functionalities of these interfaces are to be further developed after consultation with the vendors.

The configuration of the IT systems used in this study is illustrated in Fig. 3. The MBSE user client represents the user interface, the Product Data Management (PDM) server contains the PDM instance and with Teamwork Cloud by No Magic, Inc., the MBSE server provides a central repository for the storage of system models. The server-based RESTful interface allows different users to access the same model or diagram simultaneously.

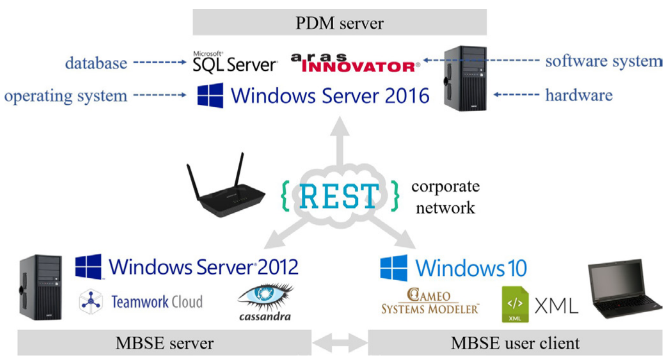

Fig. 3. Configuration of the hardware and software components used to integrate the technical system model.

3.4. Determination and implementation of interdisciplinary relationships between MBSE and BPM

The integration of system and process elements on different model levels is visualized in Fig. 4 and could be realized within the REST-based linking of IT tools. The integration is achieved by representing equivalent information objects of the domain-specific elements on the decomposed observation levels in the PDMS data model. This central administration of equivalent information objects enables in the following the direct linkage between the domain-specific elements in the PDMS.

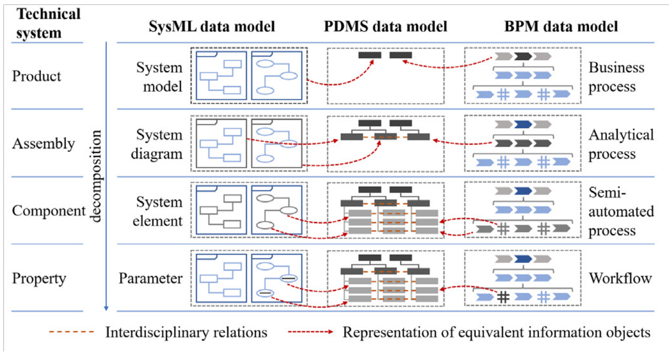

Fig. 4. Integration of equivalent information objects in a PDMS according to [28].

PDMS mainly use hierarchical structures for data management, but is also capable to map structural relationships. In the selected PDMS, this requires the design of two relationships. The first element determines the type of relationship as “interdisciplinary”. The second element is the characteristic of the “relationship” class and defines the relationship by pointing to the elements to be linked, Fig. 5.

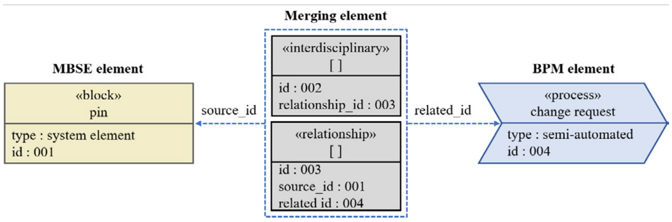

Fig. 5. Link between MBSE element and BPM element using the design of structural relations in the PDMS Aras Innovator.

In terms of systems theory, any relationship between all elements of MBSE and BPM can be mapped. In the context of this work two criteria were developed, which increase the probability of a meaningful interdisciplinary linkage of the information and system elements:

- Given reciprocal representation of a system element in the other modeling language.
- Given comparable model level of the system elements in both modeling languages.

An example of a suitable interdisciplinary linkage is the evaluation of the impact of an engineering change request

(ECR) by analyzing the SysML elements in a system diagram with a semi-automated process.

4. Use Case Scenario: Engineering Change Request

The aim of the demonstrator setup within the scope of this paper is the prototypical demonstration of feasibility for the method presented in chapter 3. One of today's challenges in product development is to master the change processes as central cost and risk factors. For this reason, the use case ECR was chosen. This use case is of high relevance for industrial applications since it considers individual customer requests and helps managing the complexity of integral products.

4.1. Use Case

In the use case, the ECR is modeled with the corresponding process steps in the PDMS Aras Innovator. By merging MBSE and BPM, it is possible to start an ECR process regarding a specific system element. The ECR can access the technical system view via the interface between BPM and MBSE. Automatically, modeled technical dependencies in the system model are then used for further execution of the ECR process. This merging is depicted abstractly in Fig. 6, using a product structure of a roller chain as use case.

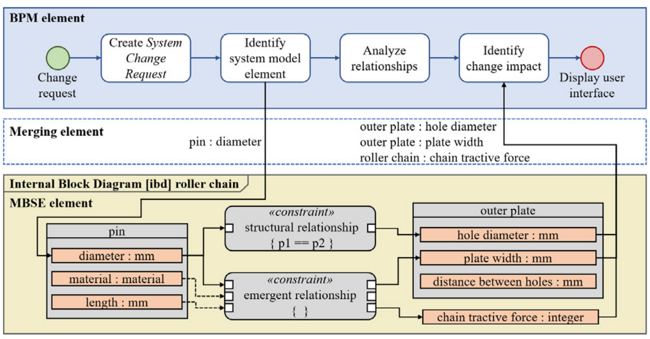

Fig. 6. Integration of MBSE elements in the simplified process model of an engineering change request.

The simplified ECR itself is visualized using BPMN and shown in the blue area of Fig. 6. The product structure together with hierarchical relationships, properties and attributes of the system modules is modeled using SysML diagrams. The regarded corresponding block elements, constraint elements and parameters are shown in the lower part of Fig. 6.

The use case in Fig. 6 shows a stakeholder triggering an ECR due to a design change of a roller chain pin. The interdisciplinary relationship (R3) between the ECR and the system model allows him to access the MBSE knowledge within the scope of his assigned activities and use it without having competence in the domain-specific modeling environment.

Within the system model, complementary relationships (R1) between the system elements of the product structure due to geometrical or physical dependencies are mapped with parametric constraints. Exemplary relationships are structural dependencies between the outer diameter of a chain pin and the drilling diameter in a chain outer plate or emergent relationships between material, length, etc. and the chain tractive force.

These relationships are automatically analyzed and fed back to the ECR as change impact. The identified change impact can be visualized in the PDMS user interface, shown in Fig. 7. If the ECR is continued, process-specific information-flows to other affected stakeholders can be automatically initiated due to transformational relationships (R2) between the process or system model and the responsible stakeholders.

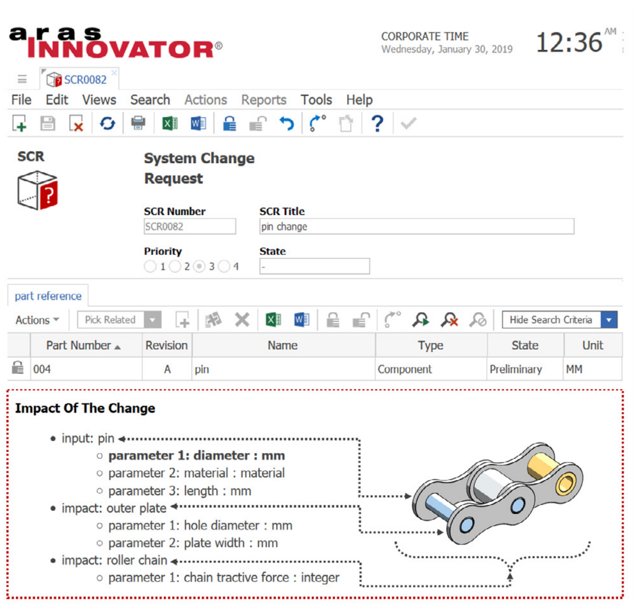

Fig. 7. Demonstration of a change request in Aras Innovator.

4.2. Use Case Discussion

In order to check the feasibility of merging MBSE and BPM, a framework concept for the integration of a PDMS platform was developed. The implementation of the procedure was evaluated within the framework of a company-specific IT infrastructure. For this purpose, various MBSE and PDMS elements on an equal observation level were merged by interdisciplinary relationships via a RESTful API. Based on a change request regarding system components, it could be shown that an integrative merging of SysML elements with process functions of the PDMS is possible. However, it must be considered that the required instantiation of the process model is limited by the integration capability of BPMN elements into a PDMS.

Based on the presented use case, the framework has been evaluated and it could be shown that the merging of BPM and MBSE with the selected software tools is feasible. The following synergy potentials were identified through the design of interdisciplinary relationships. In that way, it is possible to automatically estimate the change impact in the event of requirements. Another application of the presented strategy could be to perform design changes based on modeled domain-specific knowledge in the first place. Depending on the change impact it is possible to trigger information flows to relevant stakeholders, if necessary. The shown merging of MBSE and BPM improves the complexity management in development processes.

However, it should be noticed that the potential of merging MBSE and BPM is countered by a comparatively high initial modeling effort. With the number of considered system elements and process views on the system, the number of interdependencies to be modeled increases exponentially.

In principle, the term complexity in system theory depends on several factors and increases with element diversity and relationship diversity. According to this understanding, the reason for the high modeling effort represents its own form of complexity, which must be made controllable before a broad industrial use of MBSE in business processes is possible.

5. Conclusion

The aim of this work was to investigate the feasibility and added value of a link between MBSE and BPM. The basic prerequisite for the linkage is a modeled interdisciplinary relationship between the process model and the MBSE system model. By following the four steps of the framework presented in chapter 3, a functional link between MBSE SysML system elements and process elements in a PDMS is possible.

A demonstrator has been built to confirm the prototypical feasibility and the investigation of the synergy potentials. By merging SysML elements with a process model, it could be shown that the impact of a change in the system model can be analyzed automatically. The stakeholder triggering the ECR can access modeled system knowledge within the scope of his assigned activities. The change impact is visualized by the user interface of the PDMS and can be understood without having competence in the domain-specific modeling environment.

This combination shows high potential for Model-Based Requirements and Change Management. Despite the given feasibility, the high initial modeling effort still represents a challenge for the entrepreneurial usability of the link.

In the context of current investigations regarding the added value of merging MBSE with BPM in industrial application, it is examined how the high initial modeling expenditure can be counteracted. Statements reducing the high initial modeling effort typical for MBSE as a pure capacity problem must be critically questioned. Brute-force approaches for managing complexity through the massive use of resources have to be measurable in an entrepreneurial way and have therefore generally not proved to be assertive in the past.

References

[1] Eigner M et al. Modellbasierte virtuelle Produktentwicklung. Berlin: Springer Vieweg; 2014.
[2] Eigner M et al. An Interdisciplinary Model Based Design Approach for Developing Cybertronic Systems. In: The Design Society: DS 84: Proceedings of the DESIGN 2016 14th International Design Conference. Glasgow: Design Society, 2016, p. 1647–1656.
[3] Friedenthal S et al. A Practical Guide to SysML. The Systems Modeling Language. Amsterdam: Elsevier; 2015.
[4] Göpfert J. Modulare Produktentwicklung: Zur gemeinsamen Gestaltung von Technik und Organisation. Wiesbaden, s.l.: Deutscher Universitätsverlag; 1998.
[5] Konrad C, Löwer M, Schmidt, W, Jacobs G. Varianzsteuerung integraler Produkte durch den prozessbegleitenden Einsatz von Data-Mining Werkzeugen. In: Brökel, Grote et al. (Ed.) 2017 – Interdisziplinäre Produktentwicklung. Universität Duisburg-Essen.
[6] Gausemeier J et al. Systems Engineering in industrial practice. Survey by Hein Nixdorf Institute, University of Paderborn; wentker druck GmbH, Paderborn, 2015.
[7] Mourtzis D, Doukas M. The evolution of manufacturing systems: from craftsmanship to the era of customisation. In: Handbook of Research on Design and Management of Lean Production Systems, US, America, 2014.
[8] Mourtzis D, Fotia S, Boli N, & Vlachou E. An approach for the modelling and quantification of PSS customisation. International Journal of Production Research, 56(3), 1137-1153.Chapman, WilliamL. Engineering modeling and design. Routledge, 2018.
[9] Österle H. Entwurfstechniken, 2. Aufl. Berlin: Springer; 1995.
[10] Gadatsch A. Geschäftsprozesse analysieren und optimieren: Praxistools zur Analyse, Optimierung und Controlling von Arbeitsabläufen. Wiesbaden: Springer Vieweg; 2015.
[11] Frank U, van Laak BL. Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Institut für Wirtschaftsinformatik, Fachbereich Informatik, Universität Koblenz-Landau. Online available: http://www.wi-inf.uni-ue.de/FGFrank/documents/Arbeitsberichte_ Koblenz/Nr34.pdf, 2003, last checked 07.02.2019.
[12] Harel D, Rumpe B. Meaningful Modeling: What's the Semantics of "Semantics"? Computer 37(10), p. 64–72, 2004.
[13] Alt O. Modellbasierte Systementwicklung mit SysML. München: Carl Hanser Fachbuchverlag; 2012.
[14] Albers A et al. Model-Based Systems Engineering (MBSE) in der Karlsruher Schule: Fünf Jahre Forschung für die Anwendung. In: develop³ systems engineering; 2016. p. 38–41.
[15] Torry-Smith JM et al. Industrial Application of a Mechatronic Framework. In: The Design Society (Hrsg.): Design for X, design to X. Glasgow: Design Society; 2015. p. 247–258.
[16] Andre R, Köhler P. Wissensbasierte Unterstützung des Konstrukteurs an der Schnittstelle CAD-CAE. In: Scharr, G. (Hrsg.) et al: Tagungsband Gemeinsames Kolloquium Konstruktionstechnik: Traditio et Innovatio - Entwicklung und Konstruktion. Rostock; 2016. p. 59–68.
[17] Kößler J, Paetzold K. Support of the System Integration with Automatically generated behaviour models. In: The Design Society: Design for X, design to X. Glasgow: Design Society; 2015. p. 21–30.
[18] Mauser K et al. Taking into account the change of geometry in system simulation processes. In: The Design Society (Hrsg.): Design for X, design to X. Glasgow: Design Society, 2015 (DS / Design Society, 80,4), p. 153–162.
[19] Pinquie R et al. Property Model Methodology: A case study with Modelica. In: Horvath, I.; Pernot, J.-P.; Rusak, Z. (Hrsg.): Proceedings of TMCE 2016, 2016, p. 79–92.
[20] Motamedian B. MBSE Applicability Analysis. In: International Journal of Scientific & Engineering Research. Volume 4. Issue 2; 2013. p. 1–7.
[21] Lamm JG, Weilkiens T. Functional architectures in SysML. Proceedings of the TdSE; 2010.
[22] Weilkiens T. SYSMOD – The Systems Modeling Toolbox, Pragmatic MBSE with SysML, MBSE4U; 2016.
[23] Hooshmand Y et al. Ein integriertes Gesamtsystemmodell für die modellbasierte Entwicklung. In: Krause, D.; Paetzold, K.; Wartzack, S. (Hrsg.): Design for X: Beiträge zum 27. DfX-Symposium. Hamburg: Tutech Verlag; 2016, p. 243–254.
[24] Becker J et al. (Hrsg.). Prozessmanagement. Berlin, Heidelberg: Springer Berlin, Heidelberg; 2000.
[25] Allweyer T. BPMN 2.0 Business Process Modeling Notation. Norderstedt, Germany: Books on Demand; 2009.
[26] ISO/IEC 19510:2013. Information technology - Object Information technology - Object Management Group Business Process Model and Notation 19510:2013; 2013.
[27] Feldhusen J et al. Product Lifecycle Management für die Praxis: Ein Leitfaden zur modularen Einführung, Umsetzung und Anwendung. Berlin: Springer-Verlag; 2008.
[28] Kleinschmidt T. Integration von BPMN und UML: Wie Mars und Venus oder doch zwei Seiten einer Medaille? OBJEKTspektrum (04) 2015, p. 60–65.
[29] Tilkov S. REST und HTTP: Entwicklung und Integration nach dem Architekturstil des Web, 3. Aufl. Heidelberg; 2015.