Consistency in Cross-Generational Engineering of Cyber-Physical Systems

Albert Albers[1], Ina Schaefer[2], Lars Gesmann[1,*], Philip Ochs[2], Maximilian Fischer[1], Tobias Pett[2], Stefan Eric Schwarz[1]

[1] IPEK - Institute of Product Engineering | KIT - Karlsruhe Institute of Technology
[2] KASTEL - Institute of Information Security and Dependability| KIT - Karlsruhe Institute of Technology

** Corresponding author:*
Lars Gesmann
IPEK - Institute of Product Engineering
Kaiserstraße 10
76131 Karlsruhe
Geb. 10.23, room 808
☎ *+49 721 608 48245*
🖂 *lars.gesmann@kit.edu*

Abstract

Developing CPS is challenging: Cross-domain collaboration and cross-generational design pose fincancial risks for manufacturers, as different engineering models can become inconsistent to each other in the development process. Existing apporaches fail to ensure consistency within and across system generations. Our research addresses this issue by combining mechanical and software engineering approaches. We present a methodology using a brake system research platform to identify challenges and solutions in CPS development. The objective of our research is to formalize changes between generations of CPS and to develop algorithm-based consistency analysis methods. In the long term, we aim to facilitate cross-domain consistency management and create a systematic framework for CPS development.

Keywords

Cyber-Physical Systems, Consistency Management, Cross-Domain Engineering, Cross-Generational Engineering, Interdisciplinary Collaboration

1. Introduction

The development of Cyber-Physical Systems (CPS) is characterized by complexity due to the increasing number of subsystems, the use of different development models, advanced life cycle management, and the collaboration of different domains.

At their core, CPS combine hardware and software. Software links electronic and mechanical subsystems and enables real-time data exchange with their system environment via the Internet. The various subsystems require expertise from different domains (mechanical, electrical, and software) during the development process, making cross-domain collaboration essential. These domains work with different engineering models, such as CAD models or software architecture, with each engineering model typically providing an isolated view of the system in development.

It is essential to keep the different engineering models and their views of the system consistent because inconsistencies between development models can significantly set back the development process, resulting in time and financial risks for an organization. Ensuring consistency between engineering models is a complex task for developers.

New business models, such as over-the-air (OTA) updates or hardware upgrades, make this task even more challenging. In turn, the trend toward new business models increases the complexity of CPS engineering. On the one hand, multiple variants of a system (*variability in space*) must be offered to the market simultaneously, causing developers and companies to consider and manage those variants simultaneously, thus increasing complexity [1]. On the other hand, developers have to manage the development of several generations of a system (*variability in time*) concurrently. Even in these cases, consistency between different engineering models and their resulting views must be ensured, adding a new level of complexity and risk to the development process.

Figure 1 illustrates the evolution of a brake system from generation G_n to generation G_{n+1}, where changes in different domains lead to cross-generation inconsistencies between the engineering models involved. Therefore, managing the consistency of various engineering models across multiple versions and system generations requires a systematic and comprehensive engineering framework.

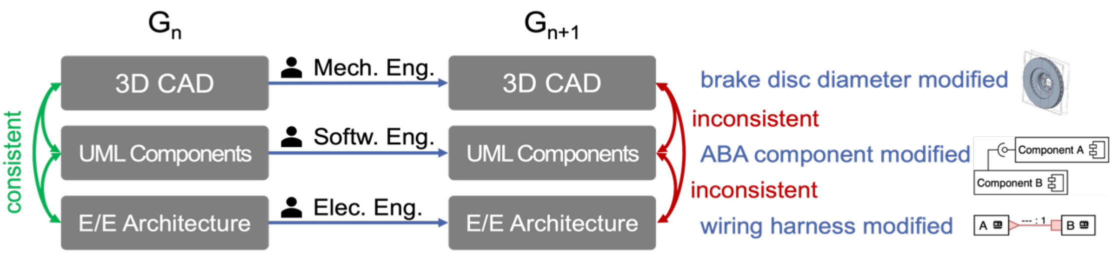

Figure 1: Cross-domain engineering can lead to inconsistencies between the models involved across system generations.

2. Current state of research

Although there are approaches on the software and hardware side that address individual aspects of the complexity of CPS engineering, these approaches have not yet been able to fully address the complexity of CPS engineering in terms of consistency management.

Some approaches attempt to address the challenges associated with cross-domain engineering by considering the entire system as a unified entity. In addition, there is a body of work from software engineering that addresses variability in space, which can be described as the challenges posed by different variants of a system. Finally, other approaches address variability in time, which refers to the challenges associated with the simultaneous development of multiple generations.

Cross-domain Engineering

In the context of cross-domain engineering, a plethora of approaches and methodologies have been developed with the aim of establishing an appropriate framework for interdisciplinary collaboration in system development.

Systems Engineering (SE) is an approach that has emerged to address the challenges of increasingly complex engineering projects. The complexity is not only due to the scope of the development process, but also to the degree of interconnectedness within the development process. Due to the relevance of different stakeholder needs, the increasing importance of life cycle management, and the growing interdisciplinarity, modern development processes can be seen as highly networked socio-technical systems. Systems engineering is a framework in which the development of a system is understood and approached as a whole [2]. This understanding is an appropriate basis for system development, but it needs to be complemented by other aspects in order to meet the challenges of CPS engineering.

Advanced Systems Engineering (ASE) serves as a guiding principle for the design of novel products and systems, striving to bridge this gap [3]. One pillar of the concept are so-called Advanced Systems (AS). The term Advanced Systems refers to the future systems that will emerge as a result of the shift from mechanical products and mechatronic solutions to intelligent, connected systems such as CPS. These systems will become increasingly complex. Systems Engineering (SE), as described above, is an approach to address complex engineering projects and to approach engineering from a holistic and interdisciplinary perspective. The final component of the ASE concept is Advanced Engineering (AE). This encompasses new techniques, processes, and methods in engineering science and practice that support engineering [4, 5]. Although the ASE concept is intended to serve as a foundation for CPS engineering, there is still a lack of concrete methods, tools, and activities to adequately support CPS engineering.

Model-Based Systems Engineering (MBSE) is one approach that falls within the scope of advanced systems engineering. MBSE supports the creation of cross-domain system models with the goal of transforming document-based, disconnected product development information into a consistent, networked system model [6]. In MBSE, the model becomes the primary artifact with the goal of representing the specification and design continuously and consistently. Various supporting techniques are available to accurately model a complex system, such as hierarchical decomposition, object-oriented modeling, graphical visualization, and visual analysis [2]. However, complex systems such as CPS are typically modeled at different levels of abstraction and from different perspectives, such as requirements, mechanical parts, state diagrams, or software architecture [6]. Ensuring consistency between these models and views is mainly based on static mappings defined by the system architects. For complex systems such as CPS, MBSE is therefore not suitable for maintaining consistency between models and views.

In the software engineering domain, the **Single Underlying Model** (SUM) is a collaborative approach that aims to keep involved engineering models consistent [7]. Therefore, formal descriptions of engineering models (meta-models) are unified into a single meta-model by identifying semantic overlaps between them. For instance, in the context of a vehicle brake system, the diameter of the brake disc is a crucial parameter in both the CAD model and the brake safety calculations. Hence, in a SUM, the brake disk diameter represents a semantic overlap between these two engineering models. However, with more and larger engineering models involved, maintaining the SUM approach tends to be challenging because the models are no longer separated as soon as they are integrated into a SUM. As a result, the SUM approach is not well-suited to the scale and complexity of modern CPS development processes.

To address the challenge of maintainability in the SUM approach, the semantic overlap between involved engineering models is virtualized by consistency relations in the **Virtual**

Single Underlying Model (V-SUM) [8, 9]. Consistency relations are formal definitions of the relationships between information in two or more models. In the case of a brake system, for instance, a consistency relation for the brake disk diameter is set up between the CAD model and the brake safety calculation. Therefore, each model can be maintained separately, and the V-SUM can be updated more easily by modifying consistency relations. However, the SUM approach and V-SUM approach do not consider multiple variants of a system (variability in space) nor multiple variants over system generations (combined variability in space and time), thus lacking a crucial part in supporting CPS engineering.

Variability in Space

Since variants exist for hardware and software – in the sense of multiple configurations of a system existing simultaneously – various approaches have been developed to address the challenges associated with the managing variants.

In product development, **variant management** refers to dealing with the variety of configurations of a product [1]. In this context, variant management is an attempt to make the inherent complexity of product development, which increases with the number of configurable elements of a product, more manageable [10]. Although variant management refers to dealing with variability in space, it does so from a different perspective. The concept is more about an understanding of product development than a concrete method or approach to support systems engineering. In addition, aspects of evolution over time are not considered at all.

Software variants can be managed as a **Software Product Line** (SPL), wherein a set of common software artifacts is customized by constrained configuration options (features and feature models) [11]. The field of SPL engineering describes processes based on roles [12], such as domain and application engineer, and techniques for theoretical analysis (problem space) and practical analysis (solution space) [13]. While there is existing work on managing generic realization artifacts in addition to software artifacts [14, 15], there is still an inherent lack of integration of cross-domain concerns into PL engineering. Furthermore, consistency specification across generations of PLs is currently limited to problem space variability modeling [16, 17].

As product line variants with many realization artifacts can get complex to assemble, **Delta Modeling** is an approach to derive product variants by defining differences (deltas) between variants [18]. A product variant is defined by a series of delta values applied to an initial variant, for example, a default baseline configuration of the system. This paradigm can be used for arbitrary formalized models, for example, a UML class model of a variant that is derived by applying a set of deltas (e.g., addition, modification, or removal of UML classes, attributes, and relations) to an initial UML class model. An extension of Delta Modeling regarding the description of evolution is given with higher-order delta modeling, as introduced below.

Variability in Time

In consideration of the variability in space, there are also various approaches from different domains for managing variability in time in order to meet the associated challenges.

Although initially proposed for variability modeling, Delta Modeling can be used to express evolution by defining deltas for changes over time instead of modifications in a variant space. A combined approach is proposed as **Higher-Order Delta Modeling**, which extends Delta Modeling to express evolution of variable systems [19]. As a variant is defined by a set of deltas, the evolution of this variant is defined by a set of changes to the set of deltas. While this concept is also applicable to arbitrary formalized models, neither Delta Modeling nor Higher-Order Delta Modeling provides a framework for ensuring consistency across multiple models.

In the context of code artifacts, **revision control systems** [20], such as Git [21], can be used to track versions in the development of software projects with one or more developers. However, revision control systems are usually tailored for code structures (e.g., line-based functionality) and thus not suitable for managing arbitrary solution space artifacts, such as CAD files.

An approach from mechanical engineering that is able to describe the evolution of systems over time is the **model of SGE - System Generation Engineering** [22]. This model provides a comprehensive description of the engineering process, employing the concept of system generations. The model is based on the assumption that the structure and subsystems of a new system generation are always based on a reference system, which defines a significant portion of the basic structure of the new system generation. These structures are either partially carried over (carry-over variation) or used as a starting point for variations [23]. While some functional units of the new system generation must adhere to a new solution principle according to the new objective (principle variation), other subsystems must be redesigned based on an existing solution principle (embodiment variation) [23]. Although the model of SGE can address the cross-generational aspect of CPS engineering, it still has some shortcomings that need to be addressed for cross-domain and cross-generational engineering of CPS [24]. For example, the application of the approach to a metamodel level remains unclear. In addition, quantified consistency management (analysis, maintenance, and repair of consistency relationships) across system generations has not yet been considered.

The objective of our research is to address the shortcomings of the existing approaches to cross-generational CPS engineering. We aim to develop an approach that ensures consistency in CPS engineering. This necessitates the consideration of cross-domain components, as well as variability in space and variability in time. The research question is thus formulated as follows:

RQ: How can cross-generational engineering of Cyber-Physical Systems (CPS) be approached to ensure consistency throughout the engineering process?

3. Methodology

Our objective is to develop a concept that combines the approaches of a V-SUM and the model of SGE. In particular, we use a V-SUM to express relationships between technical models of different domains, whereas we use the model of SGE to express relationships between generations of these models.

To support our research, we are participating in a research platform named the "Electro-Mechanical Brake System". Utilizing an automotive brake system, we aim to replicate and understand the evolution of Cyber-Physical Systems (CPS). The brake system is a sufficiently complex CPS, encompassing all relevant domains: software engineering, electrical engineering, and mechanical engineering. This research platform provides the capability to integrate additional functionalities, thereby facilitating the analysis of diverse system generations of the brake system in alignment with the model of SGE.

To answer the overarching research question from chapter 2, it is necessary to proceed sequentially and divide the research into individual sub-research questions (SRQs).

SRQ1: **How a V-SUM can be used to represent a system generation and its specific characteristics according to the model of SGE?**

For the initial sub-research question, we start to delineate the characteristics of a V-SUM for a specific system generation of a CPS. Using the research platform, we will design initial models for the system generation G_{n-1} to investigate the basic feasibility of integration and representation in a V-SUM (Figure 2, part a). Over time, additional models will follow (Figure 2, part b) to eventually capture the representation of a single system generation and its main characteristics. Simultaneously, we can use these models to investigate which inconsistencies within a system generation can be caused by changes in a model.

SRQ2: Which consistency relationships exist within and across system generations and how these can be expressed?

According to the model of SGE, the evolution of a CPS can be described using the three variation operators, that have been explained in detail in chapter 2. As each of these variation operators is characterized by a certain degree of complexity, they serve as a starting point for the analysis of inconsistencies and their respective impacts that may occur between different models. The research platform can also be used to examine this aspect. In terms of multiple system generations, the subsequent generation G_n of the "Electro-Mechanical Brake System" will be developed (Figure 2, part c). As we expand the V-SUM for system generation G_{n-1}, we investigate and analyze not only consistency relations between different models within the V-SUM of generation G_{n-1}, but also consistency relations between the two generations G_{n-1} and G_n (Figure 2, part d). This allows us to elucidate how consistency between system generations manifests and identify the consistency relationships existing between them. Additionally, we develop appropriate metrics to quantify the effects of changes to models.

SRQ3: How a V-SUM can support the maintenance of consistency within a system generation, but also across different system generations?

Following an analysis of the resulting inconsistencies of the different variation operators, it is essential to identify rules for maintaining consistency within and across system generations. To illustrate this necessity, one might consider the example illustrated in Figure 1. In this context, it is nessesary to determine the effects of an attribute variation in form of an adjusted brake disc diameter (CAD) on the necessary wiring harness components (E/E Architecture). Given the significant impact that inconsistencies resulting from a change can have, it is essential to implement a range of consistency preservation mechanisms. These can be classified as automatic, semi-automatic, or completely manual, with each type of variation operator based on the specific variation in question and its respective effect on consistency. The aim is to provide consistency preservation mechanisms based on the respective variation operator, considering different models and system generations.

Based on this foundation, our objective is to identify methods for predicting and analyzing inconsistencies resulting from changes and to suggest various design alternatives for developers. Subsequently, we intend to initially develop a manual consistency maintenance mechanism for cross-generational application in CPS development, followed by a (semi-)automated consistency maintenance mechanism. We seek to identify effective ways to represent predicted inconsistencies across models within and across system generations. For example, as shown in Figure 1, this could enable users to compare the impact of different brake disk diameters. Our vision does not only consider inconsistency within the CAD model, such as e. g. a collision checking assistance, but we want to enlarge this view across models and system generations. Ultimately, this will enhance the overall quality of decisions made by developers throughout the development process, as they will possess a comprehensive understanding of the impact of their respective decisions.

By answering the three sub-research questions previously outlined, we can address the overarching research question: "How can cross-generational engineering of Cyber-Physical Systems (CPS) be approached to ensure consistency throughout the engineering process?". The research platform provides an optimal environment for establishing a functional V-SUM. It addresses inconsistencies iteratively, encompasses all relevant domains, and facilitates the examination of diverse system generations according to the model of SGE. This makes it ideal for developing a V-SUM with all necessary functions. By involving researchers from all three domains, the identified consistency preservation mechanisms and predictive change analyses can be continuously validated. In short, the research platform enables the testing and validation of intermediate results obtained during the iterative process, ensuring high quality results.

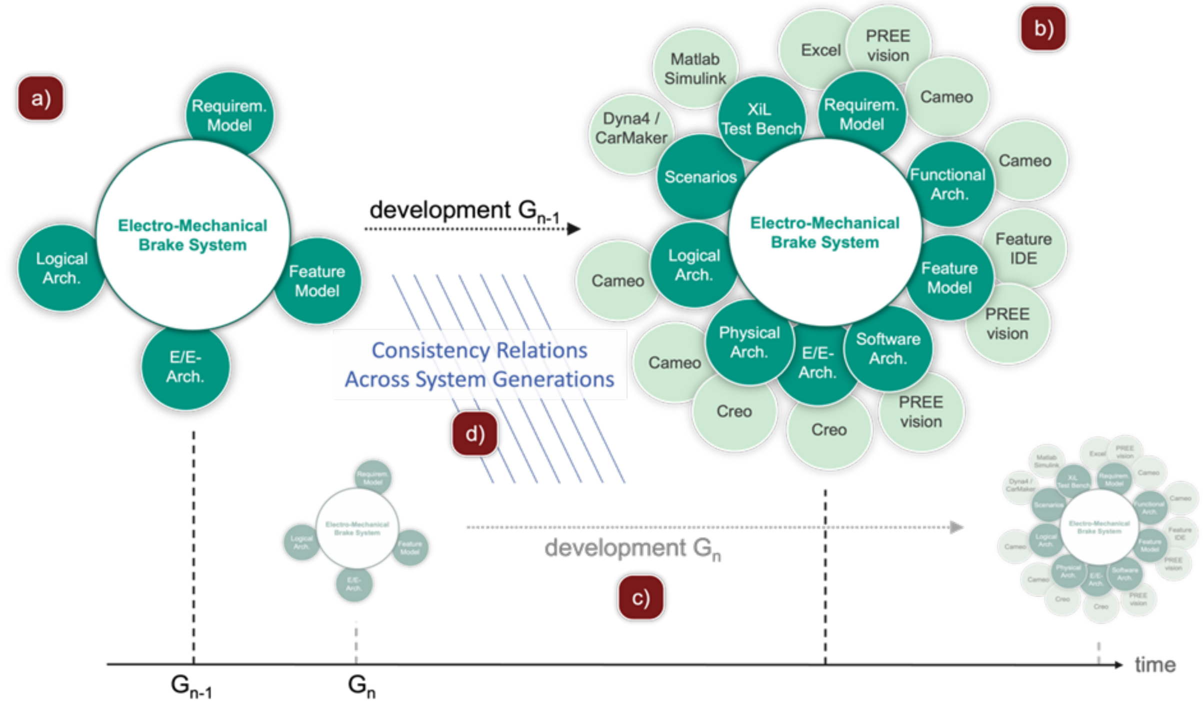

Figure 2: Using the research platform to iteratively increase complexity of the desired V-SUM.

4. Results and discussion

The objective of our research is to develop a concept that combines the V-SUM approach with the model of SGE. We believe that using a V-SUM-based approach, with its support for consistency management, is a highly promising avenue for advancing the cross-generational development of CPS in terms of system generation engineering. By combining these two approaches, we aim to maintain consistency both within a single system generation and across multiple system generations.

Specifically, our objective is to use a V-SUM to express the relationships between the technical models of different domains, while employing the model of SGE to express the relationships between the generations of these models. This will enable us to formally describe changes in development models between system generations based on the consistency descriptions of these models within a single system generation. This approach serves as a basis for consistency analysis across system generations at all stages of the development process. It ensures cross-generational system development by maintaining consistency across different generations of the technical models involved.

Building on this, we aim to evaluate the impact of design decisions within and between system generations. Since the variability of a system generation and subsequent system generations are expressed using the same mechanism, we see significant potential synergies between modeling and analysis techniques within a single system generation that can be utilized across system generations. By analyzing the impacts of changes between technical models of a system generation, we can infer inconsistencies across system generations from the resulting consistency violations. Thus, our approach provides a systematic and comprehensive framework for managing and maintaining system consistency across generations.

Using the research platform described in Chapter 3, the “Electro-Mechanical Brake System”, we will first implement the basic properties of a system generation in a V-SUM, extend the model of SGE for use in a V-SUM, and represent a single generation of the “Electro-Mechanical Brake System” in a V-SUM. Based on this, we will analyze consistency relationships within a system generation using the initial technical models. The next step is to derive consistency specifications for system generations from the findings of this analysis. On this basis, we can develop general specifications for consistency between system generations using additional generations of the “Electro-Mechanical Brake System”.

Once we have identified consistency specifications and possible causes of inconsistencies between system generations, we can establish rules for maintaining consistency. These rules will form the basis for a decision support method for design decisions. Based on planned changes, inconsistencies and their effects will be predictively detected, evaluated using risk analysis, and the best possible alternative decision will be proposed to the developer.

5. Conclusion and implications

In our research, we address consistency management issues in the development of CPS by combining approaches that are well-established in their respective fields. Our methodological approach is to design a research platform to analyze and improve the development process of a CPS across domains and generations. We aim to identify challenges and develop a suitable methodology for ensuring consistency by combining the concept of V-SUM and the model of SGE.

Consequently, our objective is to achieve cross-domain consistency management, not only within a system generation but also across system generations. This approach enables managing the increasing complexity associated with new business models and the concurrent development of multiple system variants and generations. By maintaining consistency across different engineering models and generations, we can mitigate the risks of time delays and financial setbacks, ensuring a more efficient and reliable development process.

Our approach also enables a predictive understanding of how changes to one technical model may impact others, both within the same generation and across different generations. This predictive capability, supported by risk analysis, assists developers in making informed design decisions, ultimately leading to more robust and reliable CPS.

In conclusion, the integration of V-SUM and the model of SGE provides a systematic and comprehensive framework for managing consistency in CPS development. This framework supports the development of CPS that are not only technologically advanced but also consistent and reliable across their lifecycle, from initial development through multiple generations of upgrades and changes. This research has significant implications for the future of CPS development, offering a structured approach to handling the inherent complexity and ensuring long-term consistency and reliability.

Acknowledgements

This paper is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263.

References

[1] THIEBES, Florian; PLANKERT, Nicole: Umgang mit Komplexität in der Produktentwicklung – Komplexitätsbeherrschung durch Variantenmanagement. In: SCHOENEBERG, Klaus-Peter (Hrsg.): Komplexitätsmanagement in Unternehmen. Wiesbaden: Springer Fachmedien Wiesbaden, 2014, S. 165–185

[2] ALBERS, Albert; LOHMEYER, Quentin: Advanced systems engineering - towards a model-based and human-centered methodology. In: Tools and Methods of Competitive Engineering (2012), S. 407–416

[3] ALBERS, Albert (Hrsg.); DUMITRESCU, Roman (Hrsg.); GAUSEMEIER, Jürgen (Hrsg.); LINDOW, Kai (Hrsg.); RIEDEL, Oliver (Hrsg.); STARK, Rainer (Hrsg.): Strategie Advanced Systems Engineering – Eine Leitinitiative zur Zukunft des Engineering- und Innovationsstandorts Deutschland. 2022

[4] DUMITRESCU, Roman (Hrsg.); ALBERS, Albert (Hrsg.); RIEDEL, Oliver (Hrsg.); STARK, Rainer (Hrsg.); GAUSEMEIER, Jürgen (Hrsg.): Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft: Ein Beitrag zum Advanced Systems Engineering,. Paderborn, 2021

[5] FRAUNHOFER-INSTITUT FÜR ARBEITSWIRTSCHAFT UND ORGANISATION IAO: Advanced Systems Engineering (ASE). URL https://www.iao.fraunhofer.de/de/forschung/advanced-systems-engineering.html. – Aktualisierungsdatum: 2024-06-26 – Überprüfungsdatum 2024-06-26

[6] WEILKIENS, Tim: Systems Engineering with SysML/UML: Modeling, Analysis, Design: The MK/OMG Press, 2008

[7] ATKINSON, Colin; STOLL, Dietmar; BOSTAN, Philipp: Orthographic Software Modeling: A Practical Approach to View-Based Development, Bd. 69. In: MACIASZEK, Leszek A.; GONZÁLEZ-PÉREZ, César; JABLONSKI, Stefan (Hrsg.): Evaluation of Novel Approaches to Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010 (69), S. 206–219

[8] BURGER, Erik; REUSSNER, R. (Mitarb.): Flexible Views for View-based Model-driven Development: KIT Scientific Publishing, 2014

[9] KLARE, Heiko: Building Transformation Networks for Consistent Evolution of Interrelated Models. KIT - Karlsruhe Institute of Technology, KASTEL. 2022

[10] STEGEMANN, Patrick: Integrierte Komplexitätsbeherrschung in der Produktentwicklung durch Produktkommunalitäten. UniBW Hamburg. 2010

[11] CLEMENTS, Paul; NORTHROP, Linda: Software Product Lines: Practices and Patterns. 7. print. Boston: Addison-Wesley, 2001 (SEI series in software engineering)

[12] APEL, Sven; RHEIN, Alexander von; WENDLER, Philipp; GROSLINGER, Armin; BEYER, Dirk: Strategies for product-line verification: Case studies and experiments. In: 2013 35th International Conference on Software Engineering (ICSE), S. 482–491

[13] BENAVIDES, David; SEGURA, Sergio; RUIZ-CORTÉS, Antonio: Automated analysis of feature models 20 years later: A literature review. In: Information Systems 35 (2010), Nr. 6, S. 615–636

[14] ANANIEVA, Sofia; GREINER, Sandra; KRUEGER, Jacob; LINSBAUER, Lukas; GRUENER, Sten; KEHRER, Timo; KUEHN, Thomas; SEIDL, Christoph; REUSSNER, Ralf: Unified Operations for Variability in Space and Time. In: ARCAINI, Paolo; DEVROEY, Xavier; FANTECHI, Alessandro (Hrsg.): VaMoS '22: 16th International Working Conference on Variability Modelling of Software-Intensive Systems: proceedings: Florence, Italy, February 23-25, 2022. New York, New York: The Association for Computing Machinery, 2022 (ICPS), S.1–10

[15] HENTZE, Marc; PETT, Tobias; THÜM, Thomas; SCHAEFER, Ina: Hyper Explanations for Feature-Model Defect Analysis. In: VaMoS '21: Proceedings of the 15th International Working Conference on Variability Modelling of Software-Intensive Systems 14 (2021), S. 1–9

[16] SEIDL, Christoph; SCHAEFER, Ina; AßMANN, Uwe: Integrated management of variability in space and time in software families. In: GNESI, Stefania (Hrsg.): SPLC 2014. proceedings of 18th International Software Product Line Conference. [Place of publication not identified]: ACM, 2014 (ACM International Conference Proceedings Series), S. 22–31

[17] NIEKE, Michael; SAMPAIO, Gabriela; THÜM, Thomas; SEIDL, Christoph; TEIXEIRA, Leopoldo; SCHAEFER, Ina: Guiding the evolution of product-line configurations. In: Software and Systems Modeling 21 (2022), Nr. 1, S. 225–247

[18] SCHAEFER, Ina: Variability Modelling for Model-Driven Development of Software Product Lines. In: International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS), 2010

[19] LITY, Sascha; KOWAL, Matthias; SCHAEFER, Ina: Higher-order delta modeling for software product line evolution. In: SEIDL, Christoph; TEIXEIRA, Leopoldo (Hrsg.): FOSD 2016: Proceedings of the 7th International Workshop on Feature-Oriented Software Development, 2016, S. 39–48

[20] TICHY, Walter: Design, Implementation, and Evaluation of a Revision Control System. In: Proceedings of the 6th international conference on Software engineering, 1982

[21] SPINELLIS, Diomidis: Git. In: IEEE Software 29 (2012), Nr. 3, S. 100–101

[22] ALBERS, Albert; FAHL, Joshua; HIRSCHTER, Tobias; HAAG, Steffen; HÜNEMEYER, Sebastian; STAIGER, Tim: Defining, Formulating and Modeling Product Functions in the Early Phase in the Model of PGE - Product Generation Engineering. In: 6th IEEE International Symposium on Systems Engineering (ISSE) 2020, 2020

[23] ALBERS, Albert; BURSAC, Nikola; WINTERGERST, Eike: Product generation development-importance and challenges from a design research perspective. In: New developments in mechanics and mechanical engineering: proceedings of the International Conference on Mechanical Engineering (ME 2015), 2015, S. 16–21

[24] ALBERS, A.; RAPP, S.; FAHL, J.; HIRSCHTER, T.; REVFI, S.; SCHULZ, M.; STÜRMLINGER, T.; SPADINGER, M.: Proposing a Generalized Description of Variations in Different Types of Systems by the Model of PGE – Product Generation Engineering. In: Proceedings of the Design Society: DESIGN Conference, 2020, S. 2235–2244