Product-Structuring Concepts for Automotive Platforms: A Systematic Mapping Study

Philipp Zellmer
Lennart Holsten
Volkswagen AG & Harz University
Wolfsburg & Wernigerode, Germany
philipp.zellmer2@volkswagen.de
lennart.holsten@volkswagen.de

Thomas Leich
Harz University
Wernigerode, Germany
tleich@hs-harz.de

Jacob Krüger
Eindhoven University of Technology
Eindhoven, The Netherlands
j.kruger@tue.nl

ABSTRACT

The products of the automotive industry are facing one of the biggest changes: becoming digital smart devices on wheels. Driven by the rising amount of vehicle functions, electronic control units, and software, today's vehicles are becoming cyber-physical systems that are increasingly complex and hard to manage over their life cycle. To handle these challenges, the automotive industry is adopting and integrating methods like software product-line engineering, electrics/electronics platforms, and product generation. While these concepts are widely recognized in their respective research areas and various domains, there is limited research regarding the practical effectiveness of implementing these concepts in a software-driven automotive context. In this paper, we investigate existing product-structuring concepts and methods that consider both hardware and software artifacts, and their applicability to the automotive as well as other cyber-physical industries. For this purpose, we conducted a systematic mapping study to capture a comprehensive overview of existing product-structuring concepts and methods, based on which we discuss how the state-of-the-art can or cannot help solve the challenges of the automotive industry. Specifically, we analyze the practical applicability of the existing solutions to help practitioners apply them and to guide future research.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; • **Software and its engineering → Software product lines**; **Maintaining software**.

KEYWORDS

automotive, electrics/electronics, product line, life-cycle management, cyber-physical system, product-structuring concept

ACM Reference Format:
Philipp Zellmer, Lennart Holsten, Thomas Leich, and Jacob Krüger. 2023. Product-Structuring Concepts for Automotive Platforms: A Systematic Mapping Study. In *27th ACM International Systems and Software Product Line Conference - Volume A (SPLC '23), August 28-September 1, 2023, Tokyo, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3579027.3608988

SPLC '23, August 28-September 1, 2023, Tokyo, Japan

ACM ISBN 979-8-4007-0091-0/23/08…$15.00
https://doi.org/10.1145/3579027.3608988

1 INTRODUCTION

To deliver valuable products, automotive manufacturers are constantly evolving their product portfolio by engineering innovative vehicle features and functions. Due to trends like autonomous driving, electrification, or vehicle connectivity, more and more innovations are based on software rather than hardware, leading to an increasing digitization of vehicles [7, 10, 57]. In the past, hardware platforms based on mechanical components were established within the automotive industry to build an overarching architecture for different vehicles based on reusable and cheaper components. Today, more and more software-based features must be integrated into the existing hardware platforms to fulfil new customer and legal requirements. As a consequence, vehicles have evolved into software-intensive cyber-physical systems, in which hardware and software must interact efficiently to provide innovative products.

While enabling features like over-the-air updates or self-driving capabilities, the increasing amount of software integrated into a traditional hardware platform presents challenges for automotive manufacturers. In particular, managing the variability of all (software and hardware) artifacts and their interconnections becomes increasingly difficult, resulting in disproportionately rising expenses and efforts. Consequently, automotive manufacturers demand for effective and efficient product-structuring concepts and methods that consider the vehicle as a software-centered cyber-physical system. Even though current platform strategies in practice are still focusing on mechanical components, automotive manufacturers are adopting variant-management concepts from software product-line engineering [13, 47, 60] to integrate the software perspective into their existing hardware platform strategies [19, 74]. However, designing a holistic platform strategy that consolidates all dimensions of modern vehicles remains a challenging task.

In this paper, we report a systematic mapping study with which we elicited an overview of existing product-structuring concepts that consider both hardware and software artifacts, and that are applicable to automotive as well as other cyber-physical systems. Based on the results of our study and our expertise in the automotive domain, we discuss and assess the practical applicability of the identified concepts for dealing with current challenges of automotive manufacturers. More precisely, we contribute the following:

- We review 17 papers to provide an overview of recent automotive product-structuring concepts that consider vehicles as software-intensive cyber-physical systems (Section 3).

- We extract and discuss key issues and lessons learned regarding the concepts' practical applicability (Section 4).
- We define research directions to guide the development of a practical product-structuring concept that involves software, hardware, and the interactions between both (Section 5).

Our results can guide practitioners in identifying concepts for structuring their product portfolios, and researchers in scoping new techniques for managing complex cyber-physical systems.

2 BACKGROUND AND RELATED WORK

In this section, we introduce the background of our work. Specifically, we discuss current trends in the automotive industry (Section 2.1) and product-structuring concepts (Section 2.2).

2.1 Automotive System Engineering

Automotive Innovation. Currently, the automotive industry is facing more and more requirements regarding functional safety and security, onboard communication needs, comfort, and environmental protection; resulting in an ever increasing number of vehicle features [1, 12, 78]. At the same time, software has become the key enabler for innovative features in the automotive industry, in which it has become a dominant factor for determining competition [7, 10]. Consequently, the amount of software within a vehicle has increased rapidly with nearly exponential growth each year [15, 57]. Nowadays, a significant percentage, approximately 80 % to 90 %, of innovation in the automotive industry is driven by advancements in electronics that, in turn, heavily rely on software [57]. This trend is further evident in the drastically increasing number of lines of code, with current premium cars containing over 100 million lines [15]. Although software has a great potential to facilitate and thereby reduce the costs of innovation and prototyping, vehicles are still becoming increasingly complex by integrating more and more features as well as a growing number of electronic control units (ECUs), sensors, and actuators [6, 7, 9, 11, 78]. As a result, the development costs for modern vehicles are rising and the automotive industry is encountering new challenges in managing their product portfolios comprising reused software, ECUs, numerous customizable vehicles, and various vehicle generations [14].

Platform Engineering. To efficiently develop and manage their product portfolios, automotive companies have established platform strategies as variability-management instruments [8, 69]. The basic concept of automotive platforms follows a simple idea: Instead of developing and evolving individual vehicles independently, key vehicle components are consolidated into a (hardware) platform, which is developed once and deployed into a number of vehicles, enhancing reuse and creating overarching synergies [17, 31, 56, 65, 70, 77]. Despite the increasing digitization of vehicles, automotive platforms are still dominated by mechanical vehicle components, which is why automotive companies have only recently started to adopt more and more concepts from software engineering [25]. In fact, software product lines rely on the same idea to manage software systems by integrating a number of reusable artifacts as well as their variation points into a software platform [13, 47, 60]. So, software platforms can systematically reduce the time-to-market, decrease costs, and enhance software quality by enabling reuse and standardization of software artifacts [44, 45, 68, 76]. However, automotive manufacturers struggle to consistently implement software platforms throughout their whole product portfolio, due to a still remaining high reliance on hardware platforms. In addition, hardware and software components of modern vehicles are deeply interconnected with each other and distributed between different manufacturers, demanding for more integrated engineering methodologies that consider vehicles as cyber-physical systems [32]

Cyber-Physical Systems. Cyber-physical systems are complex systems in which physical and software components are closely interconnected and interact with each other depending on their operational context and environment. They are designed to monitor and control physical devices within the system by using digital communication. [43, 49, 75, 79]. Covering an increasingly diverse range of industries, cyber-physical systems are becoming a driver for innovation, with the potential to evolve beyond today's information systems [49, 59, 67]. Cyber-physical systems are already used in a wide range of domains, including high-confidence medical devices and critical infrastructure control [49].

The automotive industry is investing significant resources in enhancing the intelligence of their vehicles and production systems through increased integration of connectivity, electronics, and software components [43, 62]. As a result, modern vehicles are evolving into complex distributed cyber-physical systems, typically involving more than a hundred heterogeneous processors, multiple interconnected subsystems with various sensors and actuators, numerous radio interfaces, and connections with other vehicles, infrastructure, or backend systems [43, 62]. This enables new functionalities, such as intelligent mobility assistances, smart home applications, and x-by-wire systems, to utilize information from vehicles, their underlying infrastructures, or backend systems [35, 48, 62].

Automotive Life Cycle Management. A product life cycle management system is a comprehensive and centralized system that oversees a product from its inception to its disposal or retirement with the goal of increasing productivity [28, 29, 73]. The life cycle of products can vary in duration and content, depending on the industry, and can be influenced by factors such as product innovation or consumer behavior [64, 81]. To achieve multi-market saturation and gain competition advantages, all industries are expanding their product portfolios by introducing more derivatives and variants, resulting in more and more electrical, electronic, and software components that enable communication between systems [38]. The resulting cyber-physical systems are much more complex to handle and need to be managed longer in their life-cycle [18, 38].

As the automotive industry becomes more dependent on software, the ability to update vehicles with new features or to fix identified issues is becoming more important to meet evolving customer needs or requirements [33, 52, 58, 81]. Due to the large number of devices, it can be time-consuming, inefficient, and troublesome to perform update processes in the traditional way, specifically through service centers [16, 26, 38]. To bypass this issue, the automotive industry is investing in over-the-air (OTA) updates, which allow for remote updates of vehicle features or bug fixes, making these updates more efficient and scalable [16, 26, 38]. As a result, OTA updates play a key role in the automotive industry to achieve customer benefits via software updates during a vehicle's life cycle.

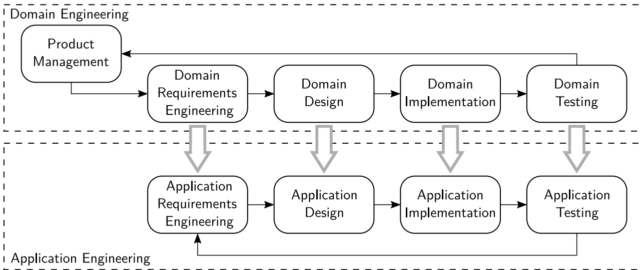

Figure 1: Software product-line engineering based on Pohl et al. [60] and adapted from Krüger [44].

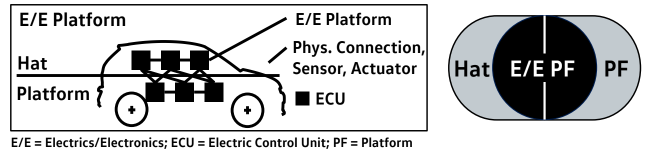

Figure 2: The electrics/electronics platform concept based on Holsten et al. [30].

In this paper, we refer to this new life cycle management within the automotive industry as "software life cycle management," which describes life cycle management via OTA software updates.

2.2 Product-Structuring Concepts

We refer to a product-structuring concept as any methodology that attempts to systematically manage a large product portfolio of related, yet customized, products. In the following, we detail typical platform strategies that are relevant for our work, but many more exist in specific domains (e.g., clone-and-own management for software variants [66]). These concepts are the most important ones for our work, because they can be integrated within a complex platform involving hardware, software, and ECUs.

Software Product-Line Engineering. Product-line engineering is based on the principle that a number of similar products share a set of core assets that can be explicitly defined and reused across these products, leading to a customizable platform. Within the product line, increased reuse and standardization create synergies between the individual product variants. Applied to software engineering, software product lines have been established as a key variability-management concept for software-intensive systems [13, 34, 50, 60]. Software product-line engineering involves two main processes: domain engineering and application engineering, as we display in Figure 1 [23, 36, 47, 60]. Domain engineering involves developing the core assets, including all software artifacts as well as their interconnections, consolidated into a software platform. In addition to the overall set of reusable artifacts, the software platform also includes the artifact's constraints as a specification for deriving concrete product variants [50, 55, 63]. Application engineering involves configuring and deriving concrete products from the software platform to fulfill specific customer requirements [50, 74]. By increasing reuse and standardization within a systematic variability management framework, software product lines promise to significantly reduce costs, increase software quality, and achieve a faster time-to-market [19, 21, 44, 45, 68, 71, 76].

Electrics/Electronics Platform Engineering. Following the same idea of establishing an integrated platform, electrics/electronics platforms have been proposed as a means for consolidating a set of common vehicle components into an overarching architecture for a number of different vehicle models. In contrast to hardware or software platforms, electrics/electronics platform engineering focuses on integrating software and hardware artifacts into an overarching electrics/electronics architecture. To illustrate this concept's application in the automotive industry, we display the electrics/electronics platform with its relations to the hardware platform and hat strategy established in this domain in Figure 2 [30, 32, 61]. Instead of differentiating between mechanical parts (hardware platform) and customer-relevant parts (hat), the electrics/electronics platform concept combines all electrics/electronics vehicle components into a single layer. This includes all software artifacts as well as their physical representations as ECUs. As an overarching connection layer, the electrics/electronics platform generates a fundamental electrics/electronics architecture that closely links software and hardware artifacts and considers the vehicle as a cyber-physical system. So, characteristic benefits of hardware platforms, such as increased reuse and overall synergies, can be optimized across all software-related vehicle components. A high degree of usability to fulfill varying demands (e.g. different equipment lines or sales markets) as well as adaptability to respond to technological developments and innovations are considered key factors for successful electrics/electronics platforms [32, 61].

Product-Generation Engineering. The concept of product-generation engineering is based on the assumption that mechatronic products in general and modern vehicles in particular are rarely developed from scratch, but rather based on an already existing product—the so-called reference product [2, 3]. A similar strategy exists in software engineering with the widely observed clone-and-own development [72]. The engineering process for a new product generation consists of adopting existing components and systems from the reference product as well as the development and integration of new subsystems. We can distinguish different types of variations: carry-over variation, attribute variation, and principle variation (cf. Figure 3). Carry-over variation describes the adoption of individual elements from the reference product, allowing for adjustments needed to satisfy interface specifications. For instance, existing technical solutions of the reference product may be applied to a new vehicle generation. The term attribute variation covers changes to particular vehicle attributes, such as adaptions to the geometrical shape of components or specific functional parameters, while the underlying technical and functional concept remains unchanged compared to the reference product. Principal variations are defined as engineering activities that add new elements or links to the reference product or that remove existing ones. Within product-generation engineering, different forms of principal variations exist, including new or adjusted vehicle features, adaptations to manufacturing processes, and additional software artifacts as well as their linkage [4]. Taking into account the different types of variations, reuse and standardization across successive vehicle generations can be enhanced. Additionally, overarching synergies

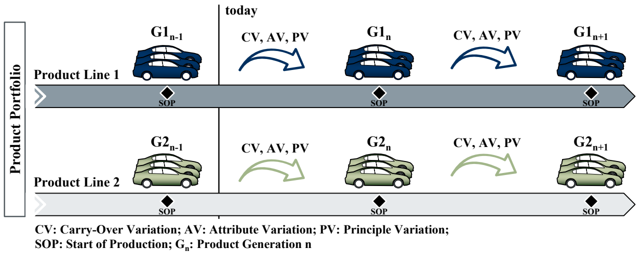

Figure 3: The product-generation engineering concept based on Albers et al. [3].

between different vehicle models can be exploited, utilizing the reference product as the basis for several products or even product lines. In recent research, product-generation engineering has been evolved to incorporate the increasing software relevance and digitization of vehicles. For this purpose, product-generation engineering is applied to vehicle functions by establishing overarching functional roadmaps that map the functional evolution across the whole product portfolio and life cycle [3, 20].

Related Work. We found three publications that overlap with our mapping study of automotive product-structuring concepts. However, none of these works provides a comprehensive and systematic mapping study that adequately addresses our research goal. Marchezan et al. [54] report a systematic literature review regarding the scope of software product lines. While the automotive domain and similar product-structuring concepts are not addressed, there are some publications that are relevant in the context to software product-line engineering. Kenner et al. [37] conducted a systematic literature review (2011–2020) focusing on safety and security concerns in product-line engineering, but the automotive domain and product-structuring concepts are again not addressed. Finally, Knieke et al. [42] describe a systematic literature review (2016–2021) that focuses on holistic approaches to manage the evolution of automotive software product-line architectures. The authors analyzed 107 studies centering around automotive software product lines, without papers related to similar product-structuring concepts. Although their study is related to our research, our emphasis is not on software product lines alone. Instead, we provide an overview of automotive product-structuring concepts including key issues and lessons learned regarding the concepts' practical applicability.

3 METHODOLOGY

We aimed to understand the current state-of-the-art on structuring product portfolios using the previously introduced concepts. For this purpose, we employed a systematic mapping study following the guidelines for software-engineering research proposed by Kitchenham [39] and Kitchenham et al. [40]. Proceeding from these guidelines, our methodology for the systematic search involves the steps we illustrate in Figure 4 and explain in this section.

3.1 Research Questions

The increasing digitization and networking of vehicles have led to new challenges, including higher complexity, demands for technical innovations, software-based vehicle maintenance, and cyber

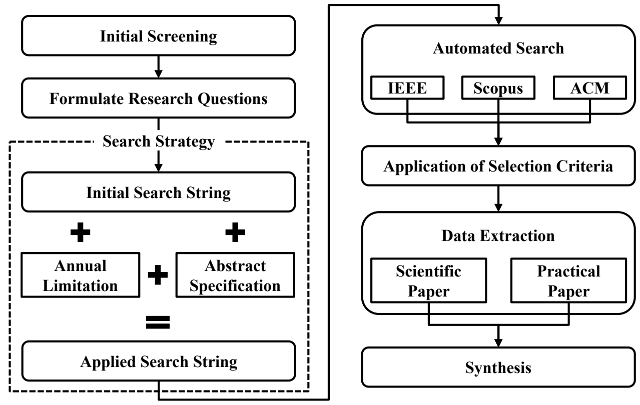

Figure 4: Overview of our research methodology.

security. To tackle these challenges, automotive companies are increasingly adopting methods from software engineering, while also considering the specific requirements of the automotive industry. In this paper, we aim to assess the practical applicability of current software-oriented concepts for structuring and managing product portfolios with respect to requirements of the automotive industry. To achieve this goal, we defined four research questions (RQs):

RQ_1 *What software-oriented product-structuring concepts exist to efficiently develop automotive-product portfolios?*
We aimed to gather an overview of concepts that help transferring today's hardware-focused automotive platforms into overarching software-related product-structuring concepts. This change in perspective helps overcome automotive challenges regarding variability and life-cycle management. In this paper, we summarize the respective literature and extract key characteristics, such as the conceptual idea and the practical applicability of the concepts.

RQ_2 *What are the challenges of applying the identified product-structuring concepts in practice?*
To answer this research question comprehensively, we examined the challenges reported in each paper in more detail. Moreover, we reflected on our practical experiences to investigate major issues and lessons learned for applying these concepts in the automotive industry.

RQ_3 *How are the concepts related to each other?*
Based on the issues and lessons learned, we compared the individual product-structuring concepts and derived commonalities as well as differences. As a result, we map the relations between the concepts and identify promising conceptual advancements and combinations.

RQ_4 *What limitations do the concepts have?*
Finally, we critically evaluated the identified concepts in terms of current automotive requirements. We discuss which directions future research should follow to solve the reported limitations and to improve the concepts' practical applicability within the automotive industry.

Tackling these research questions contributes an overview of the research landscape for practitioners and helps researchers scope new directions to solve real-world problems.

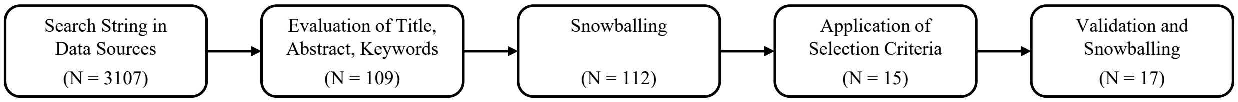

Figure 5: Stages of our search process and the number of papers we selected.

3.2 Search Strategy

According to Kitchenham and Charters [41], the selection of search terms as well as the definition of search resources are key for a systematic and reliable search strategy. To generate search strings, we defined key terms that are the basis for our search string. Reflecting on our research questions, we specified three key questions that we supplemented with synonyms and related terms: "What?", "How?", and "What for?" Each key question represented an individual search string, combining the included terms with an OR operator. Based on the individual search strings, we built the following initial search string by connecting each string with an AND operator:

> (("automo*" OR "car" OR "vehicl*") AND ("complex*" OR "varia*" OR "varie*") AND ("E/E?archite*" OR "product line" OR "E/E?develop*" OR "E/E?releas*" OR "E/E?plat*" OR "electri*/electronic*"))

We applied this search string to the full–text of papers listed in relevant databases to conduct test runs.

During the test runs, we found that the search string returned too many results, which is why we decided to limit the search results further. For this purpose, we defined the time period we considered relevant to span from 2007 until 2022. Furthermore, we complemented the search string with the following search term, which we linked to the initial search string with an AND operator and employed to abstracts only:

> ("automo*" OR "car" OR "vehicle*" OR "complex*" OR "varia*" OR "varie*" OR "E/E?archite*")

We extended and thereby improved the completeness of our search by performing backwards snowballing [46, 80]. Specifically, we analyzed the references listed in each selected primary study to identify further relevant papers that may have been missed during the database search.

We deployed our final search string to the following digital libraries:

- IEEE Xplore: Highly-cited publications in electrical engineering, computer science, and electronics.
- Scopus: Largest database of peer-reviewed papers, resulting in a strong coverage of the relevant literature.
- ACM Digital Library: Collection of full-text articles and bibliographic records covering the fields of computing and information technology.

To apply our search strategy to all of the above mentioned databases, we modified our search string in terms of special characters and research area (limiting it to computer science and engineering).

3.3 Selection Criteria

We defined the following inclusion criteria (ICs) for identifying relevant papers based on our research questions:

IC$_1$ The paper is concerned with product-structuring concepts in the automotive industry.

IC$_2$ The paper has been published between 2007, when the most relevant research has started, and 2022, the year we conducted the search.

IC$_3$ The paper has been published in a peer-reviewed journal, conference, or workshop.

IC$_4$ The paper exceeds three pages.

Moreover, we defined the following exclusion criteria (ECs):

EC$_1$ The paper is published in another language than English or German (we considered German, due to our proficiency and awareness of highly relevant work being published in it).

EC$_2$ The paper is published only as a bachelor's thesis, master's thesis, or technical report.

EC$_3$ The paper is published with incomplete or missing information about the publisher or publication type (gray literature).

Applying these selection criteria, we ensured to exclusively select papers that are aligned with our research questions and accessible to most researchers.

3.4 Data Extraction

We extracted the following standard data from each paper:

- Source
- Author(s)
- Title
- Publisher
- Publication year
- Number of pages
- Study type (i.e., scientific paper or practice report)

In addition, we synthesized specific information according to the study type to address our research questions. For *research papers* only, we collected the following additional information:

- Main concept and methods reported within the paper
- Summary of the findings and limitations
- Research objectives
- Industrial applicability of the concepts and methods

For *practice reports* (i.e., case studies) only, we extracted the following additional information:

- Business context of the case study
- Case study domain
- Main concept and methods applied within the paper
- Objectives
- Lessons learned and outlook

We carefully studied the full text of each selected paper to extract this information. In Section 4 and Section 5, we report and discuss our results, respectively.

3.5 Conduct

In Figure 5, we display the conduct of our literature search and the number of papers we ended up with after each step. We conducted

Table 1: Identified studies from each data source.

Data Source	Results	Selected		
		SPLE	E/E-PF	PGE
IEEE Xplore	1,777	2	1	1
ACM Digital Library	598	2	2	0
Scopus	732	3	1	1
Snowballing	-	0	1	3
Sum	3,107	7	5	5

the automated search in May 2022 for each data source listed in Table 1, retrieving a total of 3,107 papers. Subsequently, the first and second authors independently screened all publications to identify those that provide direct evidence relevant for our research questions. During this screening process, we applied our inclusion and exclusion criteria to guide our selection. Disagreements between the two authors were resolved through discussions until we found a common denominator. After analyzing titles and abstracts, we kept 109 papers and performed backwards snowballing on these papers, which led to the identification of three more papers. Then, we continued our review process by reading each publication in detail, which led to a set of 15 papers. Finally, we performed a cross-validation of the selected papers, extracted data, and re-iterated the snowballing (including the newly found papers), which led to a final set of 17 papers in the end (cf. Table 1).

3.6 Analysis

To analyze the extracted data, we built on our knowledge of the related work, the studies we identified, and our experiences from practice [30]. The practical experiences stem from the first two authors working in the automotive industry for several years. Both have worked in various departments of one of the largest international automotive companies, Volkswagen AG. The first author is member of a project-management team, focusing on variant management, platform engineering, and software-portfolio management. The second author is located in the life-cycle management department, focusing on software-change management and digital life-cycle management. In addition, they are both working with multiple experts in these fields, which allows them to discuss novel concepts with a broader perspective. As a result, the first two authors have a detailed understanding of current concepts related to platform engineering and (digital) life-cycle management, especially when it comes to their practical application. In this context, the authors used their expertise to formulate the criteria for evaluating the results of the mapping study as well as to analyze and assess the different concepts through subsequent discussions.

4 RESULTS

In this section, we present the main results of our mapping study. Within our literature search, we extracted a total of 17 papers, which we analyzed according to the criteria we defined for the data extraction, using three levels of fulfillment: *completely* (●), *partly* (◐), and *not* (○) fulfilled. We consider an assessment criterion to be *completely fulfilled* if it is adequately and explicitly documented

Table 2: The data we extracted from each publication.

		Mapping Perspective		Technical Perspective		Evolution		Validation			Application		
Reference	Product-Structuring Concept	Scientific	Practical	Product Perspective	Platform Perspective	Product LC	Software Update LC	Issues in Application	Lessons Learned	Decision Support	HW/SW Integration	Variability Management	Tool Support
[19]	SPL	○	●	○	●	●	○	○	○	◐	◐	●	◐
[21]	SPL	○	●	◐	◐	○	○	●	●	◐	◐	●	◐
[22]	SPL	○	●	○	●	●	○	○	◐	◐	◐	●	○
[27]	SPL	○	●	◐	◐	●	○	○	○	○	○	●	○
[53]	SPL	○	●	◐	◐	●	○	●	◐	◐	◐	●	◐
[63]	SPL	●	○	○	●	◐	○	○	○	○	●	●	●
[74]	SPL	●	○	◐	◐	◐	○	◐	○	◐	◐	●	◐
[10]	E/E-PF	●	○	◐	◐	◐	○	○	○	◐	●	●	◐
[24]	E/E-PF	●	○	○	●	●	○	●	○	◐	●	◐	◐
[25]	E/E-PF	●	○	○	●	○	○	◐	○	○	●	◐	○
[32]	E/E-PF	●	○	○	●	●	○	○	○	◐	●	●	○
[51]	E/E-PF	●	○	○	●	◐	○	◐	○	○	●	◐	◐
[2]	PGE	●	○	◐	◐	●	○	○	◐	○	◐	◐	○
[3]	PGE	○	●	○	●	●	○	●	●	◐	◐	●	○
[4]	PGE	●	○	◐	◐	◐	○	◐	◐	◐	◐	◐	◐
[5]	PGE	○	●	◐	◐	●	◐	●	●	◐	◐	●	○
[20]	PGE	●	○	○	●	●	○	◐	◐	○	●	◐	○

●: Completely fulfilled; ◐: Partly fulfilled; ○: Not fulfilled
SPL: Software Product Line; E/E-PF: Electrics/Electronics Platform;
PGE: Product-Generation Engineering; LC: Life Cycle

in a paper. To any criterion that is only insufficiently reported or examined in a different context, we assign the attribute *partially fulfilled*. Otherwise, we consider the criterion to be *not* fulfilled. We provide a complete overview of our mapping to which we refer in this section in Table 1.

4.1 Perspectives

First, we present our results regarding the different points of view adopted by the papers in terms of the research concept, the mapping perspective, and the technical perspective to outline the principal orientation of the papers we selected.

Product-Structuring Concepts. We identified the three major concepts we introduced in Section 2 to fulfill our requirements for product structuring a portfolio: software product-line engineering, electrics/electronics platform engineering and product-generation engineering. The distribution of papers between those three concepts is relatively balanced with seven papers addressing software product lines and five papers referring to electrics/electronics platforms as well as product-generation engineering each. For simplicity, we assigned one main concept to each paper, even though some papers are mentioning multiple concepts.

Mapping Perspective. Next, we distinguish between practice reports that provide insights about the concepts' practical application and scientific papers that focus on solutions on a conceptual level. As we can see in Table 1, the overall ratio between the two categories is rather balanced. Interestingly, within the software product-line category, we found mainly practice reports fulfilling our requirements (5/7), whereas we could not identify any practice reports regarding automotive electrics/electronics platforms (0/5). Despite the product-generation engineering concept being relatively new, we already found some papers (2/5) evaluating this concept's practical applicability.

Technical Perspective. As we intended to ensure with our selection criteria, the platform perspective is at least partially considered in all selected papers. Still, about half of the papers (8) combine platform and product-specific perspectives, for instance, introducing product lines as an overarching concept, but performing product-specific validation within a product line at subsystem level (e.g. Manz et al. [53]). Comparing the papers from the research fields, we can see that the tendency to focus on products is more prevalent in those regarding software product lines (4/7) and product-generation engineering (3/5), In contrast, most of the electrics/electronics platform papers (4/5) entirely adopt a platform perspective.

4.2 Evolution

Next, we analyze what evolution scenarios the individual papers cover, considering the production and maintenance of a system.

Production Life Cycle. Most papers in our dataset cover the production life cycle at least partially (15), providing support in managing software-intensive, automotive product lines throughout their whole production phase. In this context, continuous variability management (e.g., Flores et al. [22]) as well as customer-oriented innovation management (e.g., Gleirscher et al. [24]) seem to be key areas of interest. We found the strongest coverage of the production life cycle in the field of product-generation engineering, while the papers regarding electrics/electronics platforms show the lowest share of complete life cycle coverage (2/5).

Software Update Life Cycle. In contrast to the production life cycle, the software update life cycle, which is currently receiving growing attention within the automotive industry, is mentioned in only one of our selected papers [5]. We exclusively found concepts and methods improving the efficient evolution of automotive product portfolios throughout the production phase. Despite the increasing software focus, the extension of the platform idea beyond production has not yet received much attention in research.

4.3 Validation

Next, we outline validation insights reported in the papers, including practical issues, lessons learned, and concrete decision support.

Issues in the Application. The majority of papers in our dataset (10) report some kind of issues that arise in the practical application of their respective concepts. However, only five papers give a specific overview about the identified issues in a well-structured way. Analyzing the reported issues, we observed high similarities irrespective of the product-structuring concept used. Various papers emphasized missing *tool support* (6) as well as *insufficient knowledge management* due to a lack of cross-divisional communication (6) as key challenges. Besides, *inadequate model-based approaches* (5), *missing traceability* (4), as well as *overwhelming variability* (3) are reported as main issues throughout each concept. Concept-specific issues include an *insufficient end-to-end electrics/electronics architecture* focus [24, 51] as well as a *lack of functional orientation within product development* [5, 20], which hamper the practical application of the respective research concepts.

Lessons Learned. To further assess the practical application of the identified product-structuring concepts, we analyzed for every paper to what extent it provides practical guidance in terms of lessons learned. In total, we extracted lessons learned from eight papers, including only three papers explicitly addressing learnings as part of their work. Unsurprisingly, most papers reporting lessons learned have a practical background (5), while only two practice reports do not present any lessons learned. Besides, we observed major structural differences comparing the different product-structuring concept papers with each other. Regarding product-generation engineering, we found that all of the respective papers refer to practical guidance in some way, while none of the electrics/electronics platform papers discusses lessons learned to improve the concept's practical applicability. In terms of content, we identified rather consistent suggestions for the individual concepts, which, however, differ between the concepts. For software product-line engineering, several papers point out the *need for consistent cross-divisional processes* to improve knowledge management and support managing complex software systems that are influenced by a number of different organizational units—thus, focusing on optimizing variability management [21, 22, 53]. In contrast, most product-generation engineering papers propose a paradigm shift in vehicle development towards *a more functional-orientated perspective*, enhancing reuse and transparency throughout different product lines as well as life-cycle phases [3, 4, 20].

Decision Support. None of the papers we selected offers adequate support for decision makers. We identified decision-support elements in half of the papers (8). The percentage of practice reports that provide at least some sort of decision support (5/6) significantly prevails the share for scientific papers (3/11). However, in most cases the proposed support for decision makers lacks concrete guidance, including practical solutions for specific problems or issues we identified. Instead, rather generic or broad recommendations that offer minimal assistance for specific decision making prevail.

4.4 Application

Finally, we analyzed to what extent the product-structuring concepts fulfill requirements needed in the automotive domain, considering the integration of hardware and software, variability management, as well as existing tool support.

Hardware/Software Integration. Most of the selected papers recognize the necessity to integrate hardware and software artifacts into overarching product-structuring concepts to successfully apply them within the automotive industry. However, few papers consequently apply integrated concepts or models, and instead they often concentrate either on software artifacts (e.g., Fischer et al. [21] or hardware components (e.g., Albers et al. [2]). In fact, only one paper on each product-generation engineering and software product lines adopts an integrated perspective. However, every electrics/electronics platform related paper emphasizes the value of combining hardware and software artifacts in analyzing future automotive product-line concepts.

Variability Management. Each paper within our dataset provides insights to improve variability management through their respective product-structuring concepts. Unsurprisingly, managing variability plays a central role within all papers regarding software product lines. While still considering variability management, several papers from electrics/electronics platform as well as product-generation engineering focus on different topics, such as innovation management (e.g., Gleirscher et al. [24]) or life-cycle management (e.g., Albers et al. [3]). Altogether, we identify variability management as a key objective for improvement throughout the majority of our selected papers.

Tool Support. About half of the papers (9) address the application of their respective concept by extending existing tools. However, these attempts are distributed very differently among the individual concepts: While the majority of papers regarding software product lines deal with possible tool support, the tool-based implementation is addressed only once in the selected product-generation engineering papers. Additionally, we found no paper that builds and validates its own tooling within the automotive industry. Thus, the tool support for the automotive domain seems limited.

$\mathbf{RQ_1}$ & $\mathbf{RQ_2}$: Research Concepts and Practical Challenges

Software product lines, electrics/electronics platforms, and product-generation engineering are promising concepts studied in the literature to fulfill today's requirements of the automotive industry. Across all three concepts, the lack of tool support as well as insufficient knowledge management within automotive companies are reported as key challenges for the practical application.

5 DISCUSSION

In this section, we examine to what extent the different product-structuring concepts we identified in our study satisfy today's automotive requirements. For this purpose, we synthesize our results for each concept, focusing on key potentials as well as challenges for their practical application by reflecting on our experiences in the automotive domain. Finally, we compare our findings and derive future opportunities for research.

5.1 Software Product-Line Engineering

Starting with the mapping perspective, we assigned the majority of software product line papers to the practical category, which is not surprising as this concept is already established within software engineering. However, we found only few papers assessing the software product-line concept as a leading automotive product-structuring concept at platform level, replacing or extending existing hardware platforms. We noticed similar patterns in the technical perspective of the papers, since product-line engineering is rarely applied at automotive platform level, but instead employed at component or subsystem level (cf. Section 4.1). These findings support the assumption that holistically applying software-engineering concepts in the automotive industry remains a challenging task.

In this context, the inadequate documentation of variability and the lack of tool support are stated as challenges across different papers in our dataset (cf. Section 4.3). Since additional issues, such as the lack of communication between organizational units or the poor traceability throughout different process steps, are directly reducible to those main challenges, we found most papers focusing on improving the documentation of variability. Despite mentioning the missing tool support, only few papers refer to possible tools supporting the implementation of automotive software product lines (cf. Section 4.4). The reported lessons learned rather show a particular emphasis on variability-management methods and processes that are applied consistently across different organizational units as well as throughout the whole life cycle. At this point, the papers display a high degree of similarity in terms of issues and lessons learned, although we note that only a minority of the papers addresses both issues and lessons learned at all.

Our results on the evolution show that the continuous development within product life cycle frameworks is already covered for automotive use cases of software product lines. However, we found no papers investigating software product-line engineering in the context of software update life cycles regarding, for instance, OTA updates to vehicles at the customer (cf. Section 4.2). As current automotive trends like the increasing vehicle digitization lead to a growing importance of subsequent software updates, software update life cycle processes and methods could extent today‘s automotive software product-line research. To sum up, despite being widely established in software engineering, we found a number of challenges that impede the practical implementation of software product lines at vehicle platform level.

Insights: Software Product-Line Engineering

Applying software product-line engineering to automotive systems is currently limited to subsystem level, due to inadequate documentation of variability and a lack of tool support.

5.2 Electrics/Electronics Platform Engineering

Our findings indicate that the electrics/electronics platform concept fulfills key requirements of today‘s automotive manufacturers by integrating hardware and software artifacts into an overarching platform for the entire product life cycle. Despite the promising concept, we found exclusively theoretical papers regarding the electrics/electronics platform concept within our papers, which indicates that the concept has not yet been systematically applied in practice (cf. Section 4.1). Our data from the technical perspective may provide an explanation: The electrics/electronics platform concept relies on being adopted at platform level to exploit its benefits, implying that the concept must be introduced at once across various vehicle models. The current focus of automotive companies on hardware platforms as well as the long-term planning and development processes for vehicles and platforms impede a complete conceptual transition at a certain point in time. Consequently, the issues reported focus on the lack of applicability of the concept to date and identify the insufficient overarching understanding of automotive electrics/electronics architectures as well as the strong hardware focus as major obstacles.

The lack of practical experiences is also reflected in the validation parts of the selected papers, as we found little practical guidance and no papers reporting lessons learned (cf. Section 4.3). Despite acknowledging that consistent methods and tools will enhance the successful implementation of electrics/electronics platforms, we found little evidence of possible tool support in the papers (cf. Section 4.4). In summary, we see great potential in the concept to take automotive platform strategies to the next level and enable automotive manufacturers to efficiently manage today's software-intensive vehicles. However, further research is needed regarding concrete tools and methods to improve the concept's practical applicability as well as integrate the software update life cycle.

Insights: Electrics/Electronics Platform Engineering

Applying electrics/electronics platform engineering to the automotive domain seems promising, but the practical applicability has not been investigated.

5.3 Product-Generation Engineering

Product-generation engineering is the only concept in our dataset that is not based on software engineering, but originates from mechanical engineering and focuses on optimizing the development of subsequent product generations. However, in recent papers, the functional dimension including hardware and software artifacts has become the focus of attention, proposing overarching functional roadmaps that support the variability management throughout the whole product portfolio and its life cycle (cf. Section 4.2). Despite being a relatively new concept, we found a number of practice reports applying the concept in an automotive environment, demonstrating its practical feasibility (cf. Section 4.1). In this context, several issues as well as lessons learned are reported, concerning the concept's present state as well as current automotive trends (cf. Section 4.3).

Since vehicles are transitioning into cyber-physical systems, the importance of software is constantly growing. Yet, automotive companies are still mainly focusing on mechanics. The lack of functional orientation within automotive companies seems to impede the consistent application of product-generation engineering, which is why the lessons learned are focusing on enabling automotive companies to systematically plan and develop hardware and software at the functional level. Besides these industry-related issues, we found concept-specific challenges, such as the lack of model-based techniques or suitable tools, which point out the early state of the concept and which substantiate the need for future research (cf. Section 4.4). Nonetheless, we believe that the product-generation engineering concept has great potential in supporting platform strategies, as it combines the variability of hardware and software artifacts throughout the entire production life cycle based on functional roadmaps.

Insights: Product-Generation Engineering

Applying product-generation engineering in automotive companies supports efficient life cycle management, but the software oriented research on this concept is at an early stage.

5.4 Comparison and Future Research

In addition to the previous concept-specific analysis, a comparison of the results across the research concepts offers interesting points for discussion, too. The high variation within the mapping perspective indicates different levels of maturity resulting in diverging practical experiences across the concepts. Although the electrics/electronics platform concept is supposed to be an evolution of the software product-line concept, adapted to the requirements of the automotive industry, our results show that, so far, solely the application of dedicated software platforms has been evaluated.

The technical perspective displays a consistent platform orientation throughout the concepts, with software product-line and product-generation engineering papers showing tendencies towards a product-oriented application. As we discussed previously, immediate piloting of overarching concepts at product-portfolio level is challenging, which is why the concepts are typically tentatively implemented at a more limited scale. In this context, it seems reasonable that the electrics/electronics platform papers in our dataset focus on the platform perspective, since we found no practical implementation of the concept.

Unexpectedly, comparing the reported issues, we found strong similarities between the concepts. The inadequate knowledge management and the lack of tool support are unanimously identified as key challenges for the practical application of each concept. Following the distinguishing focus between the concepts, we found differing lessons learned derived from the reported issues. The software product-line papers emphasize the improvement of variability management techniques through concrete methodologies and consistent processes in response to the issues mentioned. In contrast, the product-generation engineering papers in our dataset focus on transforming automotive development processes towards systematic functional orientation to enable the concept's successful practical application. Despite reporting similar challenges, the different perspectives of the concepts lead to individual lessons learned. However, we noticed that the lessons learned are rarely translated into concrete practical guidance, including few specific recommended actions or explicit tools to implement. In fact, none of the papers across all three concepts completely fulfills our requirements for decision support. Therefore, we deduce that further practical studies are necessary to facilitate and expedite the effective implementation of the concepts in the automotive industry.

Despite the differences, our analysis demonstrates that all three concepts mostly cover the individual categories of evolution, hardware/software integration, and variability management. Particularly, software product-line research focuses on variability management, electrics/electronics platform research emphasizes the integration of software and hardware artifacts, and product-generation engineering research provides additional support to improve the automotive life-cycle management. To holistically fulfil today's automotive requirements for product-structuring concepts, we suggest combining the different concepts and optimize the overall benefits.

To provide a potential starting point based on our findings, we display the conceptual connections between the three concepts as well as current hardware-platform concepts applied in the automotive industry in Figure 6. Current automotive trends lead to companies requiring overarching product-structuring concepts more

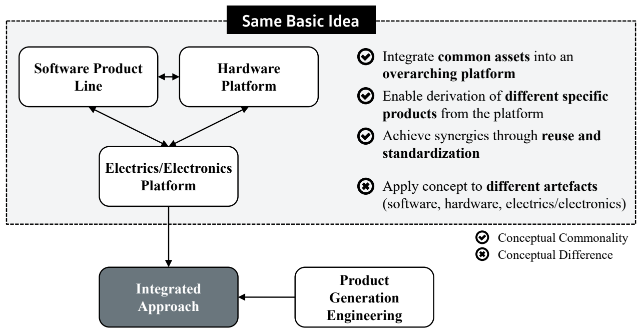

Figure 6: An overview of the conceptional connections between the concepts.

than ever to efficiently manage their extensively increasing (software) variability. In our study, we noticed that different domains emphasize varying product-structuring concepts, which, however, follow very similar conceptual ideas. Specifically, despite its differing origins and applications, software product lines, hardware platforms, and electrics/electronics platforms share the same basic platform idea, integrating common assets into an overarching platform. Based on this platform, all three concepts enable the derivation of individual products according to specific customer requirements. Besides improvements in quality and time-to-market, the three concepts focus on achieving synergy effects by enhancing reuse and standardization throughout the product line or the whole product portfolio. The main conceptual difference we perceive between the three concepts is the coverage of components and artifacts that are combined into a platform. While software product lines focus on systematic reuse across software components, hardware platforms are based on mechanic components that may or may not contain software. Electrics/electronics platforms aim at the combination of hardware and software components in the sense of functional orientation to reflect the strong interconnections of hardware and software in modern vehicles. In summary, the differences between the individual concepts appear to be reconcilable and the shared platform idea should be emphasized more as a key leverage for dealing with current automotive (variability) challenges.

Still, our results indicate that current product-structuring concepts either do not meet today's automotive requirements or have not yet been comprehensively examined and applied in automotive practice—implying further demand for research. In this context, we propose an integrated attempt: Electrics/electronics platforms seem to be most promising, combining the platform idea and representing the technical characteristics of modern vehicles best. However, this concept lacks practical applicability. In contrast, the product-generation engineering concept is designed to be embedded in a (hardware) platform-oriented environment and combines hardware and software artifacts through functional roadmaps. Therefore, we propose evaluating the combined implementation of electrics/electronics platforms and functional roadmaps as an integrated approach that optimally utilizes the overarching platform idea, incorporating additional synergies through functional orientation and guaranteeing practical applicability. Besides their value within the automotive industry we anticipate resulting concepts and methodologies to be successfully applicable in other cyber-physical domains. However, due to our practical and scientific focus on the automotive domain we are not able to give concrete practical guidelines for other cyber-physical domains.

RQ_3 & RQ_4: Connections and Limitations

Software product-line engineering, hardware platforms, and electrics/electronics platforms follow the same basic platform idea. Product-generation engineering provides additional support through functional roadmaps. Currently, the lack of practical applicability motivates future research towards an integrated perspective that unites software orientation and practical feasibility in automotive and cyber-physical domains.

5.5 Threats to Validity

We recognize that the internal and external validity of our mapping study may be compromised. First, we are aware that our search strategy does not cover all publications regarding product-structuring concepts. Our selection is based on predefined data sources and is limited to the available literature in these as well as to a selected time period. This may have introduced non-excludable biases, and thus may threaten the internal validity of our mapping study. To minimize the subjectivity, we strictly adhered to our methodology, especially the search strategy we described in Section 3.2.

Second, the level of detail regarding information we needed varied among the papers. Some describe their research in great detail, others show a lack of consistency, are ambiguous, or miss important pieces of information. We applied quality checks based on our selection criteria to mitigate such problems and built on our practical experiences to put the results into context. However, we cannot guarantee that we did not misinterpret some pieces of information.

Finally, we are aware of some threats regarding the external validity of our mapping study. As part of our search strategy, we reduced the number of papers to a final selection of 17 based on several databases, which may result in a higher potential for wrong classifications due to the small number of papers. To address this issue, we involved multiple researchers in the literature analysis and conducted an extensive literature search by using multiple databases (i.e., IEEE, Scopus, ACM) as well as snowballing. Moreover, we documented every step of our process to facilitate transparency and reproducibility.

6 CONCLUSION

In this paper, we presented a systematic mapping study on existing product-structuring concepts and methods that consider both hardware and software artifacts, and that are applicable to the automotive but also other cyber-physical systems domains. We thoroughly analyzed 17 papers covering a period of 16 years (2007–2022). Based on these papers, we identified three main product-structuring concepts: software product lines, electrics/electronics platforms, and product-generation engineering.

We found that each of these concepts has potential to align with current trends in the automotive industry. Within our analysis, we identified several issues and challenges for the practical implementation of the concepts as well as lessons learned derived from these. In this context, missing tool support and the insufficient management of knowledge within automotive companies are generally observed across all concepts. However, the derived lessons learned varied for each concept, focusing on different problem areas, such as general improvements on variability documentation or concept-specific methodological support. Connecting all our concept-specific results, we realized that software product lines, hardware platforms, and electrics/electronics platforms are based on the same basic platform idea, following highly similar conceptual ideas. For this reason, we propose an integrated perspective, combining a consolidated software-orientated platform concept and functional roadmaps building on product-generation engineering to satisfy current and future requirements in the automotive domain.

To pursue this line of research, we plan to continue with taking software-orientated automotive product-structuring concepts to the next level by engineering and validating a unified platform concept. Following the observed issues, we anticipate adequate tool support to be key to ensure the practical applicability of this concept. Since we found no paper examining the software update life cycle in-depth, it would also be important to investigate whether an integrated platform concept could help managing OTA updates.

Disclaimer. *The results, opinions, and conclusions of this paper are not necessarily those of Volkswagen AG.*

REFERENCES

[1] Heinz-Bernhard Abel, Heinrich-Jochen Blume, Ludwig Brabetz, Manfred Broy, Simon Fürst, Lothar Ganzelmeier, Jörg Helbig, Gerhard Heyen, Meike Jipp, Günther Kasties, Peter Knoll, Olaf Krieger, Roland Lachmayer, Karsten Lemmer, Wolfgang Pfaff, Thomas Scharnhorst, and Guido Schneider. 2016. *Elektrik/Elektronik/Software*. Springer.

[2] Albert Albers, Nikola Bursac, and Eike Wintergerst. 2015. Product Generation Development - Importance and Challenges from a Design Research Perspective. In *International Conference on Theoretical Mechanics and Applied Mechanics (TMAM)*. INASE.

[3] Albert Albers, Joshua Fahl, Tobias Hirschter, Marvin Endl, Rebecca Ewert, and Simon Rapp. 2020. Model of PGE – Product Generation Engineering by the Example of Autonomous Driving. *Procedia CIRP* 91 (2020).

[4] Albert Albers, Simon Rapp, Joshua Fahl, Tobias Hirschter, Sven Revfi, Micha Schulz, Tobias Stürmlinger, and Markus Spadinger. 2020. Proposing a Generalized Description of Variations in Different Types of Systems by the Model of PGE – Product Generation Engineering. *International Design Conference (DESIGN)* (2020).

[5] Masis Arslan, Fabian Haug, Nicolas Heitger, Lukas Krämer, and Albert Albers. 2016. Don't get stuck in Complexity: Coping with Strategic Complexity in the context of Product Generation Engineering. In *R&D Management Conference*. RADMA.

[6] Jakob Axelsson. 2010. A Transformation-Based Model of Evolutionary Architecting for Embedded System Product Lines. In *International Workshop on Model Based Architecting and Construction of Embedded Systems (ACES-MB)*. CEUR-WS.org.

[7] Stephan Baumgart, Xiaodi Zhang, Joakim Fröberg, and Sasikumar Punnekkat. 2014. Variability Management in Product Lines of Safety Critical Embedded Systems. In *International Conference on Embedded Systems (ICES)*. ACM.

[8] Damir Bilic, Etienne Brosse, Andrey Sadovykh, Dragos Truscan, Hugo Bruneliere, and Uwe Ryssel. 2019. An Integrated Model-Based Tool Chain for Managing Variability in Complex System Design. In *International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE.

[9] Christopher Brink, Erik Kamsties, Martin Peters, and Sabine Sachweh. 2014. On Hardware Variability and the Relation to Software Variability. In *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE.

[10] Manfred Broy. 2006. Challenges in Automotive Software Engineering. In *International Conference on Software Engineering (ICSE)*. ACM.

[11] Harald Bucher, Kevin Neubauer, and Jürgen Becker. 2019. Automated Assessment of E/E-Architecture Variants Using an Integrated Model- and Simulation-Based Approach. In *World Congress Experience (WCX)*. SAE International.

[12] Christian Buckl, Alexander Camek, Gerd Kainz, Carsten Simon, Ljubo Mercep, Hauke Stähle, and Alois Knoll. 2012. The Software Car: Building ICT Architectures for Future Electric Vehicles. In *International Electric Vehicle Conference (IEVC)*. IEEE.

[13] Paul C. Clements and Linda M. Northrop. 2001. *Software Product Lines: Practices and Patterns*. Addison-Wesley.

[14] Benjamin Cool, Christoph Knieke, Andreas Rausch, Mirco Schindler, Arthur Strasser, Martin Vogel, Oliver Brox, and Stefanie Jauns-Seyfried. 2016. From Product Architectures to a Managed Automotive Software Product Line Architecture. In *Symposium on Applied Computing (SAC)*. ACM.

[15] Yanja Dajsuren and Mark van den Brand. 2019. Automotive Software Engineering: Past, Present, and Future. In *Automotive Systems and Software Engineering*. Springer.

[16] Maria J. B. de Sousa, Luis F. G. Gonzalez, Erick M. Ferdinando, and Juliana F. Borin. 2022. Over-The-Air Firmware Update for IoT Devices on the Wild. *Internet of Things* 19 (2022).

[17] Olivier L. de Weck, Eun S. Suh, and David Chang. 2003. Product Family and Platform Portfolio Optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC)*. ASME.

[18] Martin Eigner, Walter Koch, and Christian Muggeo. 2017. *Modellbasierter Entwicklungsprozess cybertronischer Systeme: Der PLM-unterstützte Referenzentwicklungsprozess für Produkte und Produktionssysteme*. Springer.

[19] Ulrik Eklund and Håkan Gustavsson. 2013. Architecting Automotive Product Lines: Industrial Practice. *Science of Computer Programming* 78, 12 (2013).

[20] Joshua Fahl, Tobias Hirschter, Jannik Kamp, Marvin Endl, and Albert Albers. 2019. Functional Concepts in the model of PGE – Product Generation Engineering by the Example of Automotive Product Development. In *International Symposium on Systems Engineering (ISSE)*. IEEE.

[21] Stefan Fischer, Lukas Linsbauer, Roberto E. Lopez-Herrejon, Alexander Egyed, and Rudolf Ramler. 2015. Bridging the Gap between Software Variability and System Variant Management: Experiences from an Industrial Machinery Product Line. In *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE.

[22] Rick Flores, Charles Krueger, and Paul Clements. 2012. Mega-Scale Product Line Engineering at General Motors. In *International Software Product Line Conference (SPLC)*. ACM.

[23] Anilloy Frank and Eugen Brenner. 2010. Model-Based Variability Management for Complex Embedded Networks. In *International Multi-Conference on Computing in the Global Information Technology (ICCGI)*. IEEE.

[24] Mario Gleirscher, Andreas Vogelsang, and Steffen Fuhrmann. 2014. A Model-based Approach to Innovation Management of Automotive Control Systems. In *International Workshop on Software Product Management (IWSPM)*. IEEE.

[25] Sebastian Graf, Sebastian Reinhart, Michael Glaß, Jürgen Teich, and Daniel Platte. 2015. Robust Design of E/E Architecture Component Platforms. In *Design Automation Conference (DAC)*. IEEE.

[26] Subir Halder, Amrita Ghosal, and Mauro Conti. 2020. Secure Over-The-Air Software Updates in Connected Vehicles: A Survey. *Computer Networks* 178 (2020).

[27] Kengo Hayashi, Mikio Aoyama, and Keiji Kobata. 2017. Agile Tames Product Line Variability: An Agile Development Method for Multiple Product Lines of Automotive Software Systems. In *International Systems and Software Product Line Conference (SPLC)*. ACM.

[28] Mubashir Hayat and Herwig Winkler. 2022. Exploring the Basic Features and Challenges of Traditional Product Lifecycle Management Systems. In *International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE.

[29] Hannes Hick, Klaus Küpper, and Helfried Sorger. 2021. *Systems Engineering for Automotive Powertrain Development*. Springer.

[30] Lennart Holsten, Christian Frank, Jacob Krüger, and Thomas Leich. 2023. Electrics/Electronics Platforms in the Automotive Industry: Challenges and Directions for Variant-Rich Systems Engineering. In *International Working Conference on Variability Modelling of Software-Intensive Systems*. ACM.

[31] Katja Hölttä-Otto. 2005. *Modular Product Platform Design*. Ph. D. Dissertation. Helsinki University of Technology.

[32] Martin Jaensch, Bernd Hedenetz, Markus Conrath, and Klaus D. Müller-Glaser. 2010. Transfer von Prozessen des Software-Produktlinien Engineering in die Elektrik/Elektronik- Architekturentwicklung von Fahrzeugen. In *INFORMATIK*. GI.

[33] Chen Jiacheng, Zhou Haibo, Zhang Ning, Yang Peng, Gui Lin, and Shen Xuemin Sherman. 2016. Software Defined Internet of Vehicles: Architecture, Challenges and Solutions. *Journal of Communications and Information Networks* 1, 1 (2016).

[34] Eun-Young Kang, Dongrui Mu, Li Huang, and Qianqing Lan. 2017. Verification and Validation of a Cyber-Physical System in the Automotive Domain. In *International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE.

[35] Stamatis Karnouskos. 2011. Cyber-Physical Systems in the SmartGrid. In *International Conference on Industrial Informatics (INDIN)*. IEEE.

[36] Shigeo Kato and Nobuhito Yamaguchi. 2011. Variation Management for Software Product Lines with Cumulative Coverage of Feature Interactions. In *International Software Product Line Conference (SPLC)*. IEEE.

[37] Andy Kenner, Richard May, Jacob Krüger, Gunter Saake, and Thomas Leich. 2021. Safety, Security, and Configurable Software Systems: A Systematic Mapping

Study. In *International Systems and Software Product Line Conference (SPLC)*. ACM.

[38] Kevin Kerliu, Alexandra Ross, Gong Tao, Zelin Yun, Zhijie Shi, Song Han, and Shengli Zhou. 2019. Secure Over-The-Air Firmware Updates for Sensor Networks. In *International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*. IEEE.

[39] Barbara Kitchenham. 2006. Evidence-Based Software Engineering and Systematic Literature Reviews. In *International Conference on Product Focused Software Process Improvement (PROFES)*. Springer.

[40] Barbara A. Kitchenham, David Budgen, and O. Pearl Brereton. 2011. Using Mapping Studies as the Basis for Further Research – A Participant-Observer Case Study. *Information and Software Technology* 53 (2011).

[41] Barbara A. Kitchenham and Stuart Charters. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001. Keele University and Durham University.

[42] Christoph Knieke, Andreas Rausch, Mirco Schindler, Arthur Strasser, and Martin Vogel. 2022. Managed Evolution of Automotive Software Product Line Architectures: A Systematic Literature Study. *Electronics* 11 (2022).

[43] Christian F. J. König, Gerd Meisl, Natalia Balcu, Benjamin Vosseler, Henrik Hörmann, Jos Höll, and Victor Fäßler. 2018. Engineering of Cyber-Physical Systems in the Automotive Context: Case Study of a Range Prediction Assistant. In *International Symposium on Leveraging Applications of Formal Methods (ISoLA)*. Springer.

[44] Jacob Krüger. 2021. *Understanding the Re-Engineering of Variant-Rich Systems: An Empirical Work on Economics, Knowledge, Traceability, and Practices*. Ph. D. Dissertation. Otto-von-Guericke University Magdeburg.

[45] Jacob Krüger and Thorsten Berger. 2020. An Empirical Analysis of the Costs of Clone- and Platform-Oriented Software Reuse. In *Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. ACM.

[46] Jacob Krüger, Christian Lausberger, Ivonne von Nostitz-Wallwitz, Gunter Saake, and Thomas Leich. 2020. Search. Review. Repeat? An Empirical Study of Threats to Replicating SLR Searches. *Empirical Software Engineering* 25, 1 (2020).

[47] Jacob Krüger, Wardah Mahmood, and Thorsten Berger. 2020. Promote-Pl: A Round-Trip Engineering Process Model for Adopting and Evolving Product Lines. In *International Systems and Software Product Line Conference (SPLC)*. ACM.

[48] Peter Gorm Larsen, John Fitzgerald, Jim Woodcock, Peter Fritzson, Jörg Brauer, Christian Kleijn, Thierry Lecomte, Markus Pfeil, Ole Green, Stylianos Basagiannis, and Andrey Sadovykh. 2016. Integrated Tool Chain for Model-Based Design of Cyber-Physical Systems: The INTO-CPS Project. In *International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*. IEEE.

[49] Edward A. Lee. 2008. Cyber Physical Systems: Design Challenges. In *International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE.

[50] Mole Li, Lin Guan, Charles Dickerson, and Alan Grigg. 2016. Model-Based Systems Product Line Engineering with Physical Design Variability for Aircraft Systems. In *System of Systems Engineering Conference (SoSE)*. IEEE.

[51] Anders Magnusson, Leo Laine, and Johan Lindberg. 2018. Rethink EE Architecture in Automotive to Facilitate Automation, Connectivity, and Electro Mobility. In *International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. ACM.

[52] Asad Waqar Malik, Anis U. Rahman, Arsalan Ahmad, and Max Mauro Dias Santos. 2022. Over-the-Air Software-Defined Vehicle Updates Using Federated Fog Environment. *IEEE Transactions on Network and Service Management* 19, 4 (2022).

[53] Christian Manz, Michael Stupperich, and Manfred Reichert. 2013. Towards Integrated Variant Management in Global Software Engineering: An Experience Report. In *International Conference on Global Software Engineering (ICGSE)*. IEEE.

[54] Luciano Marchezan, Elder Rodrigues, Wesley K. G. Assunção, Maicon Bernardino, Fábio Paulo Basso, and João Carbonell. 2022. Software Product Line Scoping: A Systematic Literature Review. *Journal of Systems and Software* 186 (2022).

[55] Andreas Metzger and Klaus Pohl. 2014. Software Product Line Engineering and Variability Management: Achievements and Challenges. In *Future of Software Engineering (FOSE)*. ACM.

[56] Marc Meyer and Alvin Lehnerd. 1997. The Power of Product Platforms: Building Value and Cost Leadership. *Journal of Product Innovation Management* 14, 6 (1997).

[57] Patrizio Pelliccione, Eric Knauss, Rogardt Heldal, Magnus Ågren, Piergiuseppe Mallozzi, Anders Alminger, and Daniel Borgentun. 2016. A Proposal for an Automotive Architecture Framework for Volvo Cars. In *Workshop on Automotive Systems/Software Architectures (WASA)*. IEEE.

[58] Teresa Placho, Christoph Schmittner, Arndt Bonitz, and Oliver Wana. 2020. Management of Automotive Software Updates. *Microprocessors and Microsystems* 78 (2020).

[59] Dmitriy P. Plakhotnikov and Elena E. Kotova. 2021. Design and Analysis of Cyber-Physical Systems. In *Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. IEEE.

[60] Klaus Pohl, Günter Böckle, and Frank Van Der Linden. 2005. *Software Product Line Engineering*. Springer.

[61] Alexander Poth. 2009. Product Line Requirements Engineering in the Context of Process Aspects in Organizations with Various Domains. *Software Process: Improvement and Practice* 14, 6 (2009).

[62] Bikash Poudel and Arslan Munir. 2021. Design and Evaluation of a Reconfigurable ECU Architecture for Secure and Dependable Automotive CPS. *IEEE Transactions on Dependable and Secure Computing* 18, 1 (2021).

[63] Paulo Queiroz and Rosana T.V. Braga. 2014. A Critical Embedded System Product Line Model-based Approach. In *International Conference on Software Engineering and Knowledge Engineering (SEKE)*. Knowledge Systems Institute Graduate School.

[64] Ulrich Raubold. 2011. *Lebenszyklusmanagement in der Automobilindustrie*. Springer.

[65] David Robertson and Karl Ulrich. 1998. Planning for Product Platforms. *Sloan Management Review* 39, 4 (1998).

[66] Julia Rubin, Krzysztof Czarnecki, and Marsha Chechik. 2013. Managing Cloned Variants: A Framework and Experience. In *International Software Product Line Conference (SPLC)*. ACM.

[67] Ricardo G. Sanfelice. 2015. Analysis and Design of Cyber-Physical Systems: A Hybrid Control Systems Approach. In *Cyber-Physical Systems: From Theory to Practice*. CRC Press.

[68] Klaus Schmid and Martin Verlage. 2002. The Economic Impact of Product Line Adoption and Evolution. *IEEE Software* 19, 4 (2002).

[69] Günther Schuh and Michael Riesener. 2017. *Produktkomplexität managen*. Hanser.

[70] Timothy W. Simpson. 2004. Product Platform Design and Customization: Status and Promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 18, 1 (2004).

[71] Laurens Sion, Dimitri Van Landuyt, Wouter Joosen, and Gjalt de Jong. 2016. Systematic Quality Trade-off Support in the Software Product-Line Configuration Process. In *International Software Product Line Conference (SPLC)*. ACM.

[72] Ştefan Stănciulescu, Sandro Schulze, and Andrzej Wąsowski. 2015. Forked and Integrated Variants in an Open-Source Firmware Project. In *International Conference on Software Maintenance and Evolution (ICSME)*. IEEE.

[73] John Stark. 2020. *Product Lifecycle Management*. Springer.

[74] Steffen Thiel, Muhammad Ali Babar, Goetz Botterweck, and Liam O'Brien. 2009. Software Product Lines in Automotive Systems Engineering. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* 1, 1 (2009).

[75] Shafiq ur Rehman, Andrea Iannella, and Volker Gruhn. 2018. A Security Based Reference Architecture for Cyber-Physical Systems. In *Applied Informatics*. Springer.

[76] Frank Van der Linden, Klaus Schmid, and Eelco Rommes. 2007. *Software Product Lines in Action: the Best Industrial Practice in Product Line Engineering*. Springer.

[77] Thomas Vietor and Carsten Stechert. 2013. *Produktarten zur Rationalisierung des Entwicklungs- und Konstruktionsprozesses*. Springer.

[78] Peter Wallin, Stefan Johnsson, and Jakob Axelsson. 2009. Issues Related to Development of E/E Product Line Architectures in Heavy Vehicles. In *Hawaii International Conference on System Sciences (HICSS)*. IEEE.

[79] Thumeera R. Wanasinghe, Mihiran Galagedarage Don, Rajeevan Arunthavanathan, and Raymond G. Gosine. 2022. Industry 4.0 based Process Data Analytics Platform. In *Methods to Assess and Manage Process Safety in Digitalized Process System*. Elsevier.

[80] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *International Conference on Evaluation and Assessment in Software Engineering (EASE)*. ACM.

[81] Tarik Şahin, Tobias Huth, Joachim Axmann, and Thomas Vietor. 2020. A Methodology for Value-Oriented Strategic Release Planning to Provide Continuous Product Upgrading. In *International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE.