Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 136 (2025) 272–277

www.elsevier.com/locate/procedia

35th CIRP Design 2025

DPAF - Digital Platform Software Architecture Framework: Designing Software Architecture for Digital Platform Business Models linked to Cyber-Physical Systems

Jonas Höfflin, Pascal Goschnick, Patrick Brecht*, Carsten H. Hahn

[a]*Karlsruhe University of Applied Sciences, Moltkestrasse 30, 76133 Karlsruhe, Germany*

* Corresponding author. Tel.: +49-721-925-2700. *E-mail address:* patrick.brecht@h-ka.de

Abstract

The widespread adoption of Cyber-Physical Systems (CPS) in Industry 4.0, specifically in manufacturing infrastructures, and their potential as data networks create significant opportunities for digital platforms. This research introduces the DPAF - Digital Platform Architecture Framework, a software architecture framework designed to guide the integration of CPS within system engineering processes, enabling the co-design of hardware and software components alongside the Product/Digital-Platform-Business-Model CoDesign approach. Developed using the Design Research Methodology and evaluated by four groups of product engineering experts, DPAF provides developers and system engineers with a robust tool to effectively communicate, integrate, and leverage CPS for creating a software architecture for digital platforms.

Keywords: Digital Platform Business Model; Cyber-Physical System; System Engineering; Software Architecture

1. Introduction

Today's advancements are driven by simultaneous improvements in data science, Artificial Intelligence, as well as sensors, actuators, and materials, enabling the advent of new technological systems like 3D printers and augmented reality devices [1]. These developments are transforming Systems Engineering by digitizing products, systems, and processes, creating a need for an integrated product lifecycle perspective and greater emphasis on multidisciplinary product development teams [2]. This shift is expanding the understanding of mechatronic systems by integrating computational and communication capabilities that enable interaction between physical and digital components [2]. These so-called Cyber-Physical Systems (CPS), along with compatible networking components, have significant implications for digital platform business models – particularly for businesses in mechanical engineering, where the interplay between product engineering, software engineering, and business model development is crucial for generating new economic value added [3]. Achieving this requires co-designing the system and business model, ensuring product generations maintain technical feasibility, economic viability, and market relevance. This Product/Digital-Platform-Business-Model CoDesign approach facilitates the development of digital platform business models within the model of SGE – System Generation Engineering. [4]. In CPS, the integration of hardware and software, along with networking components, enables real-time data exchange across the value chain [3].

This convergence of physical and digital domains creates opportunities for digital platform business models, such as subscription-based services, on-demand production, and

10.1016/j.procir.2025.08.048

predictive maintenance. As connectivity increases, these models gain value through richer data, deeper insights, and improved collaboration [3]. However, realizing these opportunities requires robust, scalable software tailored to manage complex interactions between CPS and users, facilitate data flow, and enable real-time functionality. Developing software for digital platform business models involves not only technical design but also value creation processes that align with the dynamics of these models, as well as user interactions [6] . Current frameworks designed for digital platform business models are limited, focusing mainly on specific applications like digital twins [7–10]. A significant gap remains for manufacturing companies aiming to adopt a Product/Digital-Platform-Business-Model CoDesign approach [4]. The main challenge is aligning hardware and software engineering processes within a unified software architecture for digital platform business models, for which no comprehensive framework currently exists.

2. Theoretical Framework

2.1. Cyber-Physical Systems

Cyber-Physical Systems (CPS) are sophisticated integrations of computation, networking, and physical processes that connect to their environment via sensors and actuators, closely coordinating digital and physical elements to perform specific functions in real-time [11, 12]. They "are automated distributed systems that integrate physical reality with communication networks and computing infrastructures", making them crucial for Industry 4.0 and enabling the development of "Smart Factories" where devices interact autonomously [5]. With a potential for global digital networks [12], CPS underscores the growing role of software in engineering, supporting adaptable and secure systems in complex industrial environments [5].

2.2. Product/Digital-Platform-Business-Model CoDesign

In the context of platform business design, the CPS plays a pivotal role in "creating substantial value added in a multisided market" [3]. Within the model of SGE - Systems Generation Engineering, the development of each system builds upon elements of a reference system [13]. These development procedures at the subsystem level, which may or may not be part of immediate market offerings, are consistently adaptations based on variations within the reference system [14, 15]. In this context, Albers [3] defines CPS and digital platform business models as enablers of the dynamic interaction among providers, customers, users, and partners by serving as intermediaries and connecting flexible and compatible networking components. To systematically develop these business models, methodologies like the SPDS - Smart Platform Design Sprint guide teams through essential phases for building robust digital platform business models [4]. Central to Albers's definition of a digital platform business model is the data integration between CPS and the digital platform business model, which enhances trust, efficiency, and economic viability for all stakeholders. Another key characteristic is network effects, where the business model's value grows as more users join, creating a positive feedback loop [3]. Network effects are crucial for scalability, ensuring a continuous flow of adaptive information, that supports robust system planning, strategic product portfolio management and alignment between software and hardware teams for seamless integration of engineering processes [4].

2.3. Software Architecture

Software architecture defines the fundamental structure of software, organizing systems, component relationships, and core design principles at a high-level of abstraction [16]. It bridges business requirements and technical implementation, significantly influencing the success of complex software projects [17]. Software architecture requires strategic decisions about structure, behavior, and interactions, impacting software quality and attributes, like scalability, performance, security, and maintainability [18]. At its core, it serves as a blueprint, guiding development and stakeholder communication [19].

Architecture differs from design: while architecture focuses on the high-level structure, design dives into the specifics of implementation within the architectural framework, a distinction essential for managing complexity in large-scale systems [17]. Software architecture synthesizes application and system layers, defining the core structure and organization of software systems, including deployment and operational needs [20]. To communicate and document software architecture effectively, modeling techniques such as the Unified Modeling Language provide contextual, interactive, structural, and behavioral perspectives via activity, use case, sequence, class, and state diagrams [21]. Architectural patterns like Model-View-Controller, layered architecture, and microservices offer established solutions to recurring design problems and must align with project requirements [17, 18, 22]. Architectural design balances trade-offs between flexibility, control, complexity and proven solutions, guided by functional and non-functional requirements with an emphasis on interoperability, security, and reliability in CPS contexts [17, 20]. Frameworks and models aid in the architectural design process. Frameworks like The Open Group Architectural Framework (TOGAF) provide methodologies for managing enterprise architectures [23], while models like C4 offer specific approaches for visualizing software architectures at various abstraction levels [24]. In reviewing existing literature on software architecture design frameworks for digital platform business models that integrate CPS, a significant gap emerges. Current frameworks often lack comprehensive approaches that simultaneously address the technical, business, and operational requirements needed to develop adaptable software architectures capable of evolving alongside CPS hardware components.

3. Research Design

Fig. 1 illustrates the applied research design in the study. This research followed the Design Research Methodology (DRM) developed by Blessing and Chakrabarti [25]. The study employed research type 3 from DRM, which included two literature-based phases applied in the Research Clarification

(RC) and Descriptive Study I (DS I), followed by a comprehensive Prescriptive Study (PS). An initial phase in the Descriptive Study II (DS II) concludes the research. The process begins with a comprehensive literature review to understand the current research landscape and identify the problem. Next, an additional thorough review of the literature in DS I to outline the present context. The PS phase involved formulating an artifact using case studies and scientific discussions. Finally, the initial DS II evaluated the framework through an expert workshop and survey.

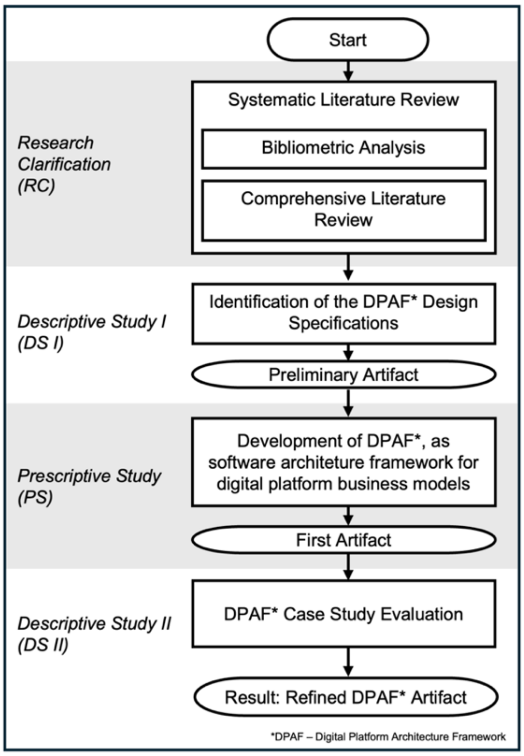

Fig. 1 Research Process
based on the Design Research Methodology (DRM) [25]

3.1. Systematic Literature Analysis

The systematic literature analysis was conducted in two phases: RC and DS I. The search strategy utilized databases including Scopus, Google Scholar, and Semantic Scholar. Search terms were organized into three broader focus areas: "digital platforms", "CPS", and "software architecture", with related terms and synonyms.

A logbook was used as the primary tool to document and organize the literature search systematically. The search covered literature in English and German published between 2014 and 2024. The bibliometric analysis, using VOSViewer [26], revealed significant findings: digital platform showed strong connections to digital transformation and Industry 4.0, while CPS had only five significant connections, with its strongest link to embedded systems. Notably, architecture frameworks only connected with software architecture, indicating a research gap in frameworks specifically designed for digital platforms integrating CPS.

The literature review found no existing architecture framework built specifically for digital platform business models integrating CPS. While frameworks like those by Petnga and Austin [8] focus on CPS architecture, they do not address digital platform integration. Notable contributions like Nölle et al. and McKee discuss digital twin platform software architectures and focus primarily on structural aspects [9, 10].

3.2. Application Evaluation

The application evaluation consisted of case studies [27], an expert workshop, and a follow-up survey. Two case studies were conducted: one regarding Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs) and another on Vorwerk's Cookidoo platform. The expert workshop involved eight participants with diverse software architecture-related backgrounds. The participants were divided into groups and tasked with designing the same digital platform software architecture using different architectural frameworks: the developed DPAF - Digital Platform Architecture Framework, C4 Model, 4+1 View Model, and arc42. A survey followed the workshop, aiming to validate and generalize the data collected. The questionnaire was structured into four dimensions: general information, framework utilization effectiveness, architecture evaluation, and metamodel assessment. Responses were collected using online questionnaires created via Google Forms, employing various question types including Likert scales and multiple-choice questions [28–30].

4. Results

4.1. DPAF - Digital Platform Architecture Framework

The DPAF follows a six-step process:
(1) define the digital platform software architecture profile, (2) outline the software architecture, (3) implement the software for the digital platform business model, (4) conduct testing of the software, (5) deploy the digital platform business model software, and (6) continue with the maintenance of the software to refine for the next iteration.

In more detail, in step (1) the *digital platform architecture profile* serves as a strategic tool for capturing and organizing critical contextual information and requirements. It comprises several key sections, as shown in Fig. 2 Fig. 3:

1. *Platform scope and vision:* Defines the core objectives and long-term goals of the digital platform, setting strategic direction for architecture.
2. *Digital platform entities:* Identifies primary stakeholders and ensures the architecture supports all user functionalities and access controls.
3. *Market and technological landscape:* Assesses current and emerging trends to guide design choices that ensure compliance and leverage new technologies.
4. *Existing CPS and systems:* Details current CPS and services, supporting integration with the platform software.
5. *Current CPS interactions:* Describes existing user and system interactions with CPS, guiding expected platform functionality.

6. *User interface and interactions:* Specifies user interactions with the platform, highlighting key functionalities.
7. *Information flow:* Maps data pathways to optimize data processing efficiency and accuracy.
8. *Communication and integration capabilities:* Defines communication types and integration needs with external systems.
9. *Connection types and protocols:* Outlines necessary connection types and security protocols regarding the CPS.
10. *Data persistence:* Provides insights into data storage needs and strategies.
11. *CPS representation:* Methods and models for the digital representation within the digital platform software.
12. *Other requirements or constraints*: Addresses additional constraints to ensure all critical factors are covered

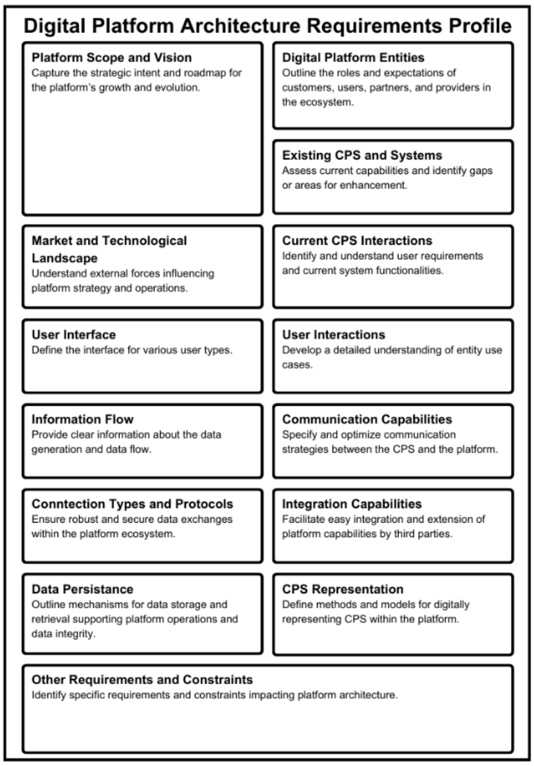

Fig. 2 Digital Platform Architecture Profile

The second step (2) the software *architecture definition* process follows a structured yet flexible top-down approach, inspired by the C4 model and specifically adapted for CPS integration. This process moves from overarching digital platform business-level down to individual software components, as shown in Fig. 3:

1. *Digital platform view*:
   This high-level view presents the digital platform software as a "black box", focusing on its interactions with external entities such as CPS, other systems, and users. It helps stakeholders understand the key components and their external relationships, often using use case or platform diagrams.
2. *Digital platform systems view*:
   In this view, the digital platform software is decomposed into its constituent systems or subsystems, showing interactions within and with external systems. It clarifies each system's role and collaboration, typically using system diagrams or the first layer of the C4 model.
3. *Flexible detailed view*:
   This view allows selective, in-depth analysis of specific systems, examining data flow, state management, or particular functionalities. It supports varying levels of detail for different parts of the architectural representation.

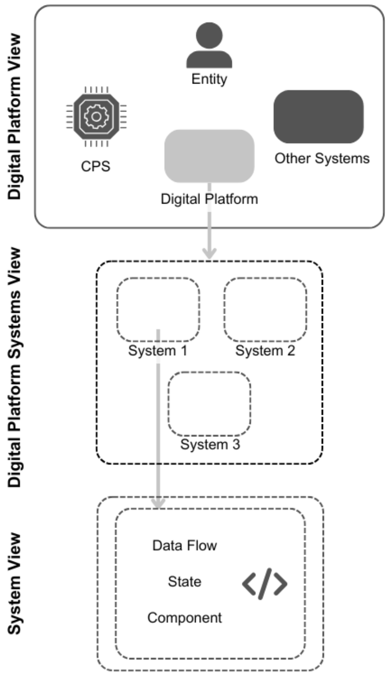

Fig. 3 Three Layers of Digital Platform Software Architecture

The DPAF - Digital Platform Architecture Framework emphasizes an iterative approach, allowing requirements set to be refined during the *architecture definition* process as new insights emerge. This flexibility is crucial for CPS integration, where adjustments between physical and digital components are often needed to ensure alignment.

In the third step (3), *digital platform Implementation*, the software is developed according to the defined architecture. The fourth step (4), *digital platform testing*, verifies functionality, performance, and security through simulations and real-world scenarios, ensuring smooth CPS integration and identifying issues early to reduce risks before deployment.

The fifth step (5), *digital platform deployment*, transitions the software from development to production. This step includes configuring the necessary infrastructure and ensuring a seamless setup in the live environment.

The sixth step (6), *digital platform maintenance*, is an ongoing process of refining the software based on user feedback and evolving requirements to prepare for the next iteration. This approach ensures that architects can focus on key system aspects while remaining adaptable to project-specific needs and the dynamic nature of CPS integration.

4.2. Evaluation of the DPAF

The DPAF underwent rigorous evaluation through an expert workshop and subsequent survey, comparing it with the frameworks C4, 4+1, and arc42. The evaluation involved eight experts with deep knowledge of CPS and software engineering. The DPAF demonstrated strengths in addressing CPS-specific requirements, scoring highest (3.67 out of 5) in meeting project requirements compared to C4 (3.0), arc42 (2.83), and 4+1 (2.67). Experts particularly valued the digital platform software architecture requirements profile and its iterative approach. The framework excelled in capturing business model aspects (4.25 out of 5) and CPS integration considerations, areas where other frameworks showed limitations. The evaluation revealed that while C4 and DPAF achieved similar completion levels (3.0), DPAF provided better context through its profiles.

4.3. Limitations and Challenges

The DPAF - Digital Platform Architecture Framework was developed to meet the requirements of digital platform business models linked to CPS, which naturally limits its broader applicability to other, more general software architecture contexts. Despite this specialized scope, the evaluation process—limited to a small group of experts and specific case studies—may not fully capture all potential use cases. Experts also noted that the framework's specificity might require additional effort for adaptation, and that some profile sections showed overlap that could be optimized, while others left gaps when handling more complex CPS integrations. In fact, the case studies showed that architecture requirements differ significantly depending on the type of CPS being integrated. Furthermore, reusing existing subsystems presented challenges, particularly in adapting legacy systems to align with modern software standards. Experts also identified a key challenge in synchronizing hardware and software development cycles, as mismatches could disrupt product timelines and impact project cohesion.

5. Discussion

This research aimed to provide a solution for companies operating within the model of SGE - System Generation Engineering to better align their software architecture with the planning needs of CPS integration and digital platform business models. Unlike existing software architecture frameworks, the developed DPAF was designed to support the Product/Digital-Platform-Business-Model CoDesign of CPS and digital platform business models. The following sections will discuss the framework's capacity to map this approach, as well as its flexibility.

5.1. Product/Digital-Platform-Business-Model CoDesign

Aligned CPS and software development is central for co-designing digital platform business model, hardware, and software within the model of SGE - Systems Generation Engineering. The DPAF was designed to bridge the gap between existing generic frameworks and the specific needs of companies that produce hardware-based products and now require integrated software solutions. A significant strength of the DPAF is its structured approach, which allows for iterative development and adaptation of both hardware and software, as well as building on established methodologies within Product/Digital-Platform-Business-Model CoDesign, such as SPDS – Smart Platform Design Sprint, which develops a digital platform business model concept that can be implemented within the DPAF.

The case studies demonstrated the DPAF's ability to map variations in digital platform business model elements and their impact on software architecture requirements, effectively supporting system modifications and functional adjustments. Findings also revealed a need for enhanced DPAF guidelines to better integrate existing CPS legacy systems with new digital components, especially given the challenge of synchronizing hardware and software development cycles to achieve Product/Digital-Platform-Business-Model CoDesign. While aligning these cycles may be difficult due to differing timelines, resource constraints, and technological limits, expert feedback suggests the framework would benefit from guidance on managing dependencies between software updates and hardware evolution. Addressing these dependencies would help companies better coordinate development cycles, reduce delays, and ensure smoother integration with digital environments—crucial for maintaining a competitive edge.

5.2. Flexible Architecture Profiles

One perceptive insight from the evaluation phase was the need for flexible architecture profiles capable of adapting to the diverse requirements of CPS integrations. Although the DPAF's iterative approach is designed to capture essential characteristics of both CPS and digital platform business model limitations emerged. Experts noted that, while efficient, the current structure lacks the complexity needed to map architecture requirements for certain types of CPS, underscoring that a one-size-fits-all approach may not be effective and highlighting the need for a more adaptable framework to handle diverse integration scenarios and CPS types. While the DPAF currently allows for some customization, further refinement is needed to develop architecture profiles that are adaptable to different industry contexts without sacrificing ease of use. This flexibility might be especially important as companies seek to leverage the full potential of CPS and digital platform business models in a coordinated and scalable manner.

6. Limitations & Future Research

This research offers valuable initial validation of the DPAF - Digital Platform Architecture Framework for CPS through case studies and expert feedback. However, the scope of empirical testing was limited which may have constrained the diversity of insights. Future studies could enhance validation by incorporating multi-case studies with interdisciplinary teams, providing a broader assessment across various CPS types. While this research focused primarily on design aspects, future studies should address practical implementation challenges, particularly the CPS and third-party system integration, to evaluate the framework's robustness and adaptability in more complex industrial environments. Another area for future research lies in refining further architecture profiles within the DPAF, balancing adaptability with detail to manage diverse CPS integrations and expanding applicability across a broader range of use cases. Additionally, the rapid pace of CPS and digital technology advancements underscores the need for regular updates to maintain framework relevance, with long-term evaluations recommended across various industries.

7. Conclusion

With this research, the DPAF - Digital Platform Architecture Framework was developed within the model of SGE - System Generation Engineering to address the specialized needs of Product/Digital-Platform-Business-Model CoDesign, integrating requirements for both digital platform business models and CPS within software architecture. Especially for companies in the "German Mittelstand", the DPAF fills a critical gap, enabling them to design software for digital platform business models alongside successive product generations of CPS. By aligning software architecture with evolving product requirements, the DPAF provides a solid foundation for Product/Digital-Platform-Business-Model CoDesign, supporting more coordinated and accelerated development cycles while enhancing the adaptability of digital solutions – ultimately helping these companies remain competitive in a rapidly changing technological landscape.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT, an AI language model by OpenAI, in order to assist in refining text for clarity and precision. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] Törngren, M., and Grogan, P., 2018, "How to Deal with the Complexity of Future Cyber-Physical Systems?," Designs, 2, p. 40. https://doi.org/10.3390/designs2040040.

[2] Eigner, M., 2021, *System Lifecycle Management*, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-62183-7.

[3] Albers, A., H. Hahn, C., Brecht, P., Cetinkaya, Ü., Pfaff, F., Schlegel, M., and Fischer, M., 2023, "Digital Platform Business Model within the Model of SGE - System Generation Engineering: Definition and Classification in Cyber-Physical Systems," *Proceedings of the International Conference on Industrial Engineering and Operations Management*, IEOM Society International. https://doi.org/10.46254/au02.20230078.

[4] Brecht, P., Keller, S., Nievera, M., Hahn, C., Pfaff, F., and Albers, A., 2023, "Product Digital-Platform-Business Co-Design: A Systematic Sprint Approach," Procedia CIRP, 119, pp. 495–500. https://doi.org/10.1016/j.procir.2023.03.111.

[5] Pivoto, D., Fernandes, L., Righi, R., Rodrigues, J., Lugli, A., and Alberti, A., 2020, "Cyber-Physical Systems Architectures for Industrial Internet of Things Applications in Industry 4.0: A Literature Review," Journal of Manufacturing Systems, 58, p. 176. https://doi.org/10.1016/j.jmsy.2020.11.017.

[6] Hein, A., Weking, J., Schreieck, M., Wiesche, M., Böhm, M., and Krcmar, H., 2019, "Value Co-Creation Practices in Business-to-Business Platform Ecosystems," Electron Markets, 29(3), pp. 503–518. https://doi.org/10.1007/s12525-019-00337-y.

[7] 2017, *Industrie4.0 Basiswissen RAMI4.0 Referenzarchitekturmodell mit Industrie4.0-Komponente*, Beuth Verlag GmbH VDE Verlag GmbH, Berlin, Wien.

[8] Petnga, L., and Austin, M., 2014, *Semantic Platforms for Cyber-Physical Systems*. https://doi.org/10.13140/2.1.2141.8245.

[9] McKee, D., 2023, "Platform Stack Architectural Framework: An Introductory Guide."

[10] Nölle, C., Arteaga, A., Egia, J., Salis, A., De Luca, G., and Holzknecht, N., 2022, "Digital Twin-Enabled Application Architecture for the Process Industry:," *Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics*, SCITEPRESS - Science and Technology Publications, Valletta, Malta, pp. 255–266. https://doi.org/10.5220/0011561800003329.

[11] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K., 2016, "Cyber-Physical Systems in Manufacturing," CIRP Annals, 65(2), pp. 621–641. https://doi.org/10.1016/j.cirp.2016.06.005.

[12] Eigner, M., Gerhardt, F., Gilz, T., and Nem, F., 2012, *Informationstechnologie Für Ingenieure*. https://doi.org/10.1007/978-3-642-24893-1.

[13] Albers, A., Rapp, S., Spadinger, M., Richter, T., Birk, C., Marthaler, F., Heimicke, J., Kurtz, V., and Wessels, H., 2019, "The Reference System in the Model of PGE: Proposing a Generalized Description of Reference Products and Their Interrelations," Proc. Int. Conf. Eng. Des., **1**(1), pp. 1693–1702. https://doi.org/10.1017/dsi.2019.175.

[14] Albers, A., Kürten, C., Rapp, S., Birk, C., Hünemeyer, S., and Kempf, C., 2022, *SGE – Systemgenerationsentwicklung : Analyse Und Zusammenhänge von Entwicklungspfaden in Der Produktentstehung*, Karlsruher Institut für Technologie (KIT).

[15] Albers, A., and Rapp, S., 2022, "Model of SGE: System Generation Engineering as Basis for Structured Planning and Management of Development," *Design Methodology for Future Products*, Springer International Publishing, Cham, pp. 27–46.

[16] IEEE, 2022, "ISO/ IEC/IEEE International Standard - Software, Systems and Enterprise Architecture Description."

[17] Richards, M., and Ford, N., 2020, *Fundamentals of Software Architecture: An Engineering Approach*, O'Reilly Media, Inc, Sebastopol, CA.

[18] Sommerville, I., 2011, *Software Engineering*, Pearson, Boston.

[19] Philip, A., Afolabi, B., Adeniran, O., Ishaya, G., and Oluwatolani, O., 2010, "Software Architecture and Methodology as a Tool for Efficient Software Engineering Process: A Critical Appraisal," JSEA, 03(10), pp. 933–938. https://doi.org/10.4236/jsea.2010.310110.

[20] Brown, S., 2015, "Software Architecture for Developers," Leanpub.

[21] Booch, G., Rumbaugh, J., and Jacobson, I., 1999, *The Unified Modeling Language User Guide*, Addison-Wesley, Reading (Mass.).

[22] Gamma, E., ed., 2011, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Boston, Mass. Munich.

[23] The Open Group, "The TOGAF® Standard — Digital Edition - Introduction." [Online]. Available: https://pubs.opengroup.org/togaf-standard/. [Accessed: 11-Apr-2024].

[24] Brown, S., 2022, "The C4 Model for Visualising Software Architecture."

[25] Blessing, L. T. M., and Chakrabarti, A., 2009, *DRM, a Design Research Methodology*, Springer London, London. https://doi.org/10.1007/978-1-84882-587-1.

[26] 2024, "VOSviewer - Visualizing Scientific Landscapes." [Online]. Available: https://www.vosviewer.com/.

[27] Schögel, M., and Tomczak, T., 2009, "Fallstudie," *Empirische Mastertechniken: Eine anwendungsorientierte Einführung für die Marketing- und Managementforschung*, C. Baumgarth, M. Eisend, and H. Evanschitzky, eds., Gabler Verlag, Wiesbaden, pp. 79–105. https://doi.org/10.1007/978-3-8349-8278-0_3.

[28] Kelle, U., Reith, F., and Metje, B., 2017, "Empirische Forschungsmethoden," *Lehrer-Schüler-Interaktion: Inhaltsfelder, Forschungsperspektiven und methodische Zugänge*, M.K.W. Schweer, ed., Springer Fachmedien, Wiesbaden, pp. 27–63. https://doi.org/10.1007/978-3-658-15083-9_2.

[29] Arning, S., and Küttel, B., 2017, "Leitfaden für die Planung, Durchführung und Auswertung einer standardisierten Befragung."

[30] Mayer, H. O., 2012, "Interview und schriftliche Befragung: Grundlagen und Methoden empirischer Sozialforschung," *Interview und schriftliche Befragung*, Oldenbourg Wissenschaftsverlag. https://doi.org/10.1524/9783486717624.