Developing Digital Twins for Smart Product-Service Systems: A Methodical Approach Demonstrated with a Fuel Cell Use Case

Renaud Kenfack[1, *], Sonika Gogineni[1], Claudio Geisert[1]

[1] Fraunhofer Institute for Production Systems and Design technology IPK

* Corresponding author:
Renaud Kenfack
Fraunhofer Institute for Production Systems and Design technology IPK
Pascalstraße 8-9
10587 Berlin
☎ *+49 30 39006 506*
🖂 *renaud.kenfack.zangue@ipk.fraunhofer.de*

Abstract

Integrating digital technologies into traditional products and services is revolutionizing various industries. Smart product-service systems (SPSS) combine smart products with data-driven services with, enhancing benefits for companies. Digital twins (DT) offer a promising approach to further enhance the value creation of SPSS, enabling observation, analysis, visualization, simulation, interaction, and integration throughout the entire product lifecycle. Existing literature lacks methodologies for SPSS that integrate DT. This paper addresses this research gap by proposing a methodology, applying a model-based systems engineering approach, as emphasized in the literature on DT and SPSS. Its effectiveness was demonstrated through the development of an SPSS for a fuel cell system.

Keywords

Smart Product-Service Systems, Digital Twin, MBSE, Smart Services, Development Methodology

1. Introduction and motivation

The rapid advancement of digital technologies and capabilities has significantly influenced various aspects of industry and economy, leading to the emergence of smart product-service systems (SPSS). They integrate smart products with data-driven services to enhance functionality and user experience [1, 2]. Digital twins (DT) present a promising concept, enabling continuous observation, analysis, visualization, simulation, interaction, and integration throughout the entire lifecycle of products. Thus, DTs offer a variety of value creation opportunities in the context of SPSS [2]. Durão et al. conducted a literature review on the current state of research regarding the integration of SPSS and DTs [3]. The key challenged they identified is the lack of development methodologies for SPSS that integrate DTs (DT-SPSS) [4]. This paper aims to address this gap and contribute to the advancement of DT-SPSS. Given that model-based systems engineering (MBSE) is frequently employed in the development of SPSS and DTs, this aspect is also be examined in this work [5, 6].

This paper is structured as follows: *Section 2* explains the fundamental concepts relevant to understanding the identified research gap. *Section 3* outlines the research objectives. *Section 4* presents the proposed methodology and its implementation. *Section 5* details the results of the implementation. *Section 6* critically assesses the effectiveness of the methodology. Finally, *section 7* summarizes the paper and provides an outlook for future research.

2. State of the art and research

2.1. Key elements of smart product-service systems

To comprehensively understand SPSS, it is essential to distinguish them from traditional product-service systems (PSS). Traditional PSS combine physical products and services to fulfill customer needs. In contrast, SPSS integrate smart products and services, characterized by embedded sensors, connectivity, and intelligent algorithms. This enables a higher degree of digital servitization [7, 8]. The concept of smart products is rooted in cyber-physical systems (CPS), which enhance mechatronic products through embedded systems and communication capabilities (e.g.: Bluetooth, OPC UA, Wi-Fi) [1, 9]. CPS involve a combination of hardware and software designed for specific tasks, with typical hardware components including processors, memory, and I/O devices, and software tailored to the system's functions [9].

Smart products extend CPS capabilities by incorporating internet-based communication, allowing them to interact with other systems and respond autonomously to environmental changes [1]. These smart services, facilitated by the Internet of Things (IoT), utilize remote services, and optimization of production processes [10]. The IoT serves as a crucial enabling technology, integrating data from smart products and platforms to offer sophisticated, application-specific solutions across various industries.

2.2. Digital twins in the context of smart product-service systems

2.2.1. Digital twin definition

The literature presents various definitions of digital twins, each emphasizing different of their intended use. [11]. Stark et al. define a DT as "*a digital representation of an active, unique product (real device, object, machine, service, or intangible asset) or unique product-service system that comprises its selected characteristics, properties, conditions, and behaviors by means of models, information, and data within a single or even across multiple lifecycle phases*" [2]. For DTs, the products in question are often CPS, which can collect operational and usage-specific data. This data forms the digital shadow of a product. By integrating the digital shadow of a product with product description models stored in the digital master, the DT is created (see Figure 1).

2.2.2. Value creation potential of digital twins

A DT serves not only as a comprehensive model of physical products but also as a system that interconnects data between physical products and their virtual counterparts. This facilitates the efficient execution of product lifecycle activities such as design, manufacturing, and service, addressing the future needs of product management across the entire lifecycle. Therefore, DT represent an effective tool for product lifecycle management [12]. The broad scope of applications holds significant potential for digital value creation in SPSS [13]. The DT is a standalone digital product with its own lifecycle. Figure 1 visualizes the interaction between the DT lifecycle and the product lifecycle (PLC) of its physical counterpart.

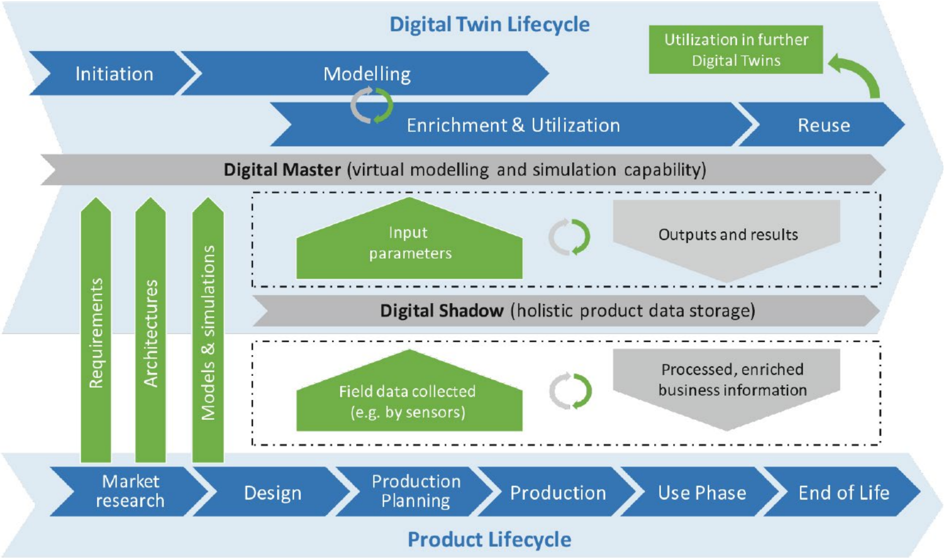


Figure 1: Relationship between the lifecycle of the product and its digital [2]

It is important to emphasize that the DT does not necessarily need to be used in every phase of the product lifecycle. Rather, the focus is on utilizing the DT economically and technically feasibly throughout the entire PLC [11]. Lindow identified various application scenarios where DTs can enhance value creation throughout the product lifecycle (PLC), highlighting their supportive role in the entire PLC [14]:

- **Begin of Life (BoL):** Optimization of product development by creating a DT for a Physical prototype.
- **Mid of Life (MoL):** Continuous monitoring and analysis, predictive maintenance, formulation of maintenance and repair measures.
- **End of Life (EoL):** Ongoing evaluation between continued use and discontinuation of a product

2.3. Development methodologies for smart-product service systems and digital twins

Existing literature includes several reviews on approaches and methodologies for the development of SPSS and DTs [15]. Halstenberg proposed a methodology defining smart-circular PSS system architectures (MESSIAH), which uses a MBSE approach for product and service architecture development [5]. Other work model-based PSS framework by Apostolov et al., which emphasizes the application of MBSE in SPSS development [6]. Other works mention that concurrent product and service development is crucial in the context of SPSS [16]. Gogineni et al. conducted a fit-gap analysis to identify suitable development methodologies for customizable IoT products, also known as smart products [17]. They found that the V-model is well suited for the DT development [18]. This conclusion was also reached by Riedelsheimer et al., which proposed a methodology for energy-efficient, customizable IoT-based smart products. The approaches discussed in this section provide insights into SPSS development methodologies. However, the current literature lacks systematic approaches specifically for developing SPSS integrating DT, underscoring the need for a structured methodology [4].

2.4. Model-based systems engineering for managing development complexity

Systems engineering is a widely recognized approach for managing complexity, offering a systematic and holistic approach to product development [19]. This approach emphasizes system thinking, which involves viewing a system as a whole and understanding the relationships between its parts, rather than isolating individual disciplines. This method captures the complexity of a system and finds more effective solutions by considering problems in their entirety [20]. MBSE extends this concept by using formalized modeling to support all development phases – problem analysis, design, implementation, verification, and validation. It involves creating and documenting detailed digital models that represent all relevant system aspects, with these models collectively forming a virtual system model. It facilitates cross-domain linkages with software tools not only during the development but also throughout later lifecycle phases, reducing the risk of errors and ensuring the system's overall sustainable functionality [21]. Existing literature discusses leveraging the system model in a digital twin's digital master to enhance its robustness throughout its whole lifecycle by serving as a single source of truth [12].

3. Research goals

To address the lack of development methodologies for DT-SPSS, this paper presents an approach for converting a physical product into a smart product while concurrently creating its DT and smart services for an SPSS. Additionally, it explores how MBSE approaches can manage the complexity in multidisciplinary system development, specifically within the context of DT-SPSS. The focus is on the phases of problem analysis, planning, and system design. The implementation of a DT-SPSS for Horizon Educational's fuel cell system (FCS) *Edustak Junior FCSU-32* will serve to evaluate the effectiveness of the proposed approach.

4. Methodical implementation of the smart fuel cell system

As discussed, the V-model is a suitable approach for developing SPSS, encompassing problem analysis, system design, domain-specific design, and system integration. This comprehensive framework makes it a widely used process model for MBSE [22]. MBSE methods, however, vary in their choice of modeling languages [19]. Many methods (e.g., MagicGrid, SysMod, OOSEM) use SysML as modeling language [23, 24]. Although SysML-based methods effectively integrate the entire development process, they require a dedicated software-based MBSE tool. Given the scope of this research, the authors selected the MBSE approach known as *Conceptual design specification for engineering complex systems* (CONSENS), which focuses on the design of systems. The eponymous modeling language is less complex than SysML and does not require a dedicated MBSE tool, which makes this method suitable for development teams lacking SysML expertise or access to MBSE tools [19].

The following subsections provide a detailed description of the methodological steps involved in the proposed approach for the mentioned application example, with a focus on scope definition and system design. As highlighted in section 2.3, the coordinated planning of the product and its services is crucial for the development of SPSS and is conducted during the scope definition phase, as illustrated in figure 2. The figure also demonstrates how CONSENS divides system design into an analysis and synthesis phase, incorporating the outlined steps.

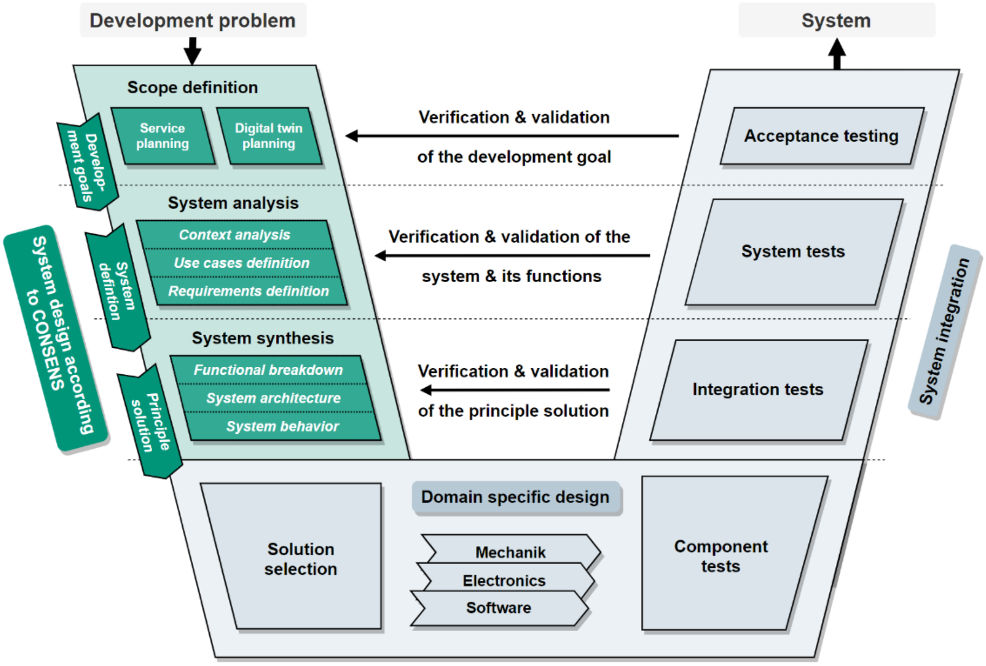


Figure 2: The proposed methodology for the development DT-SPSS

4.1. Development problem and scope definition

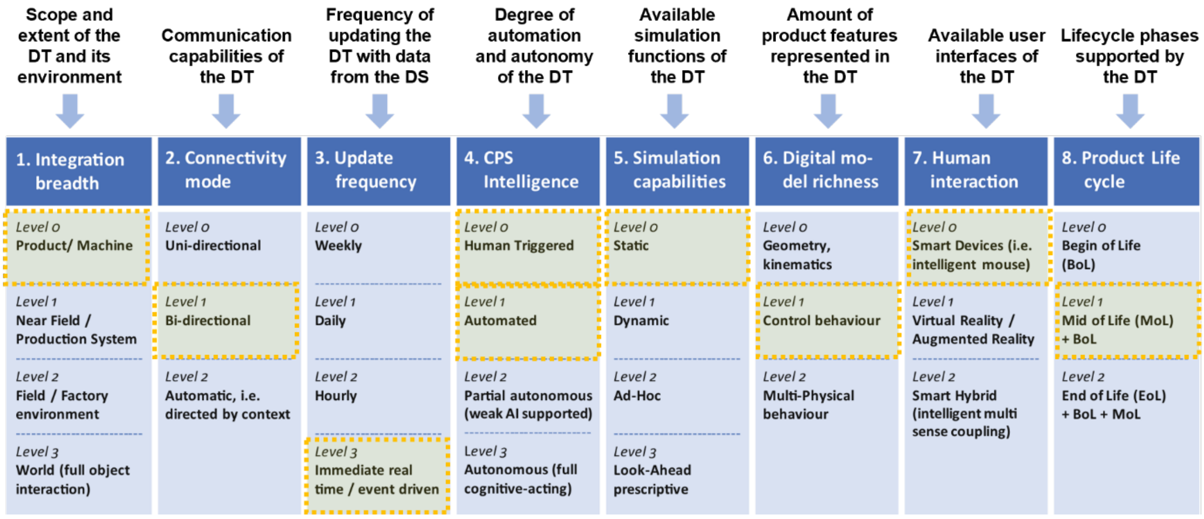
This development problem serves as the initial input for the scope definition process, which involves detailed planning of the digital twin and its associated SPSS services. This planning phase ultimately establishes the foundation for the DT-SPSS. In the considered case the development problem was to create a DT for a SPSS, specifically the *Smart Fuel Cell System* (SFCS), with the objective of optimizing fuel cell performance. Firstly, the aspired services were selected. Koldewey et al. identified 420 smart services through literature and market review, deriving twenty generic functionalities. Based on the later, the authors have chosen to implement the following services: *monitoring, alerting*, *analytics, and remote control*.

To define the scope and capabilities of the DT based on selected services, the 8D-model for DTs by Stark et al. was employed [2]. This model, encompassing eight dimensions, aids in classifying and planning digital twins, addressing aspects like integration breadth, update frequency, connectivity, CPS intelligence, simulation capabilities, digital model richness, human interaction, and lifecycle phases. Each dimension has multiple levels, where higher levels represent different realization spaces rather than superiority, though some dimensions reflect increasing richness (dimensions 2 and 7) or scope (dimensions 1 and 8) at higher levels. Figure 3 shows the selected levels of the digital twin for the SFCS across each dimension, along with their corresponding definitions.

According to the development problem, the integration breadth of the digital twin was restricted to the context of the product, specifically the fuel cell (dimension 1). The implementation of smart services required bidirectional and real-time data transmission between the product and the DT platform (dimension 2 & 3). The CPS intelligence was both automated, via service logic on the platform, and human-triggered (dimension 4), with responses occurring physically through a smart device and virtually via the platform (dimension 7). Simulation capabilities are defined as ‘static’, meaning the digital twin does not perform simulations (dimension 5). This choice reflects the research's focus on methodology rather than developing a highly autonomous DT. Since the fuel cell is an existing product, which was not developed independently, no models from the development phase (BoL) were available. The DT was designed to represent and influence the fuel cell's behavior during the MoL (dimensions 6 & 8). Considering the specified scope, the digital twin can be classified as an informative digital twin with additional user interaction capabilities according to Wilking et al. [11].

The authors selected the open-source IoT platform ThingsBoard for the DT, leveraging its capabilities for data collection, processing, visualization, device management, and user interaction. As a platform-as-a-service (PaaS) cloud solution, it reduces the implementation effort for IT infrastructure and digital capabilities. Since the fuel cell lacks communication capabilities, a *Smart Module* (SM) was developed to enable bidirectional interaction with ThingsBoard. The development goals can be summarized as follows:

- Development of a module for converting a fuel cell into a smart product (smartification)
- Software implementation of the selected smart services on ThingsBoard
- Creation of a DT dashboard on ThingsBoard for user interaction with the fuel cell



Scope and extent of the DT and its environment	Communication capabilities of the DT	Frequency of updating the DT with data from the DS	Degree of automation and autonomy of the DT	Available simulation functions of the DT	Amount of product features represented in the DT	Available user interfaces of the DT	Lifecycle phases supported by the DT
1. Integration breadth	**2. Connectivity mode**	**3. Update frequency**	**4. CPS Intelligence**	**5. Simulation capabilities**	**6. Digital model richness**	**7. Human interaction**	**8. Product Life cycle**
Level 0 Product/ Machine	*Level 0* Uni-directional	*Level 0* Weekly	*Level 0* Human Triggered	*Level 0* Static	*Level 0* Geometry, kinematics	*Level 0* Smart Devices (i.e. intelligent mouse)	*Level 0* Begin of Life (BoL)
Level 1 Near Field / Production System	*Level 1* Bi-directional	*Level 1* Daily	*Level 1* Automated	*Level 1* Dynamic	*Level 1* Control behaviour	*Level 1* Virtual Reality / Augmented Reality	*Level 1* Mid of Life (MoL) + BoL
Level 2 Field / Factory environment	*Level 2* Automatic, i.e. directed by context	*Level 2* Hourly	*Level 2* Partial autonomous (weak AI supported)	*Level 2* Ad-Hoc	*Level 2* Multi-Physical behaviour	*Level 2* Smart Hybrid (intelligent multi sense coupling)	*Level 2* End of Life (EoL) + BoL + MoL
Level 3 World (full object interaction)		*Level 3* Immediate real time / event driven	*Level 3* Autonomous (full cognitive-acting)	*Level 3* Look-Ahead prescriptive			

Figure 3: Defined scope of the SFCS according to the 8D-model for the digital twins ([2], edited)

4.2. System design according to CONSENS

Context analysis: In this step the system boundary of SFCS was defined by considering the system-of-interest (SoI) as a black-box and representing it within its environment. In this process, the environment elements are identified and their interactions with the system are examined. Figure 4 shows the developed context diagram of the SFCS.

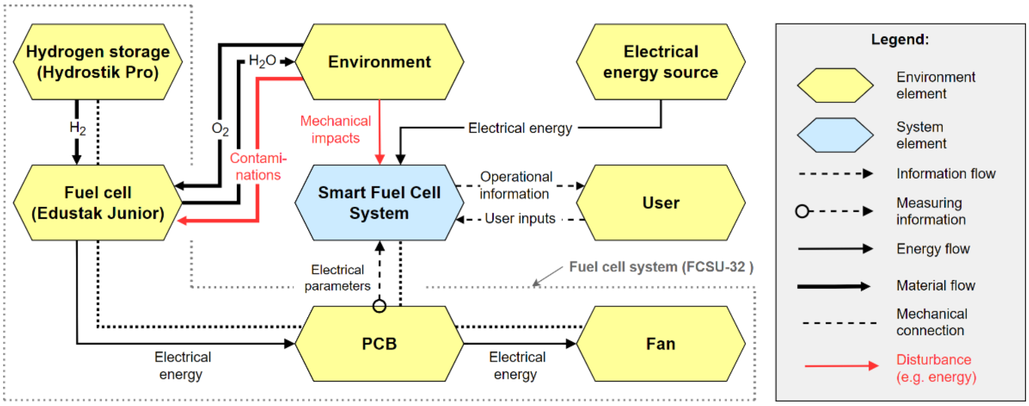


Figure 4: Context diagram

Use cases: Use cases define the application scenarios of the SoI and specify how it should behave in these scenarios. In this process, the SoI is treated as a black box to avoid premature decisions regarding the system architecture. The use cases (UC) of the SFCS, where ideated on basis of the selected smart services. Table 1 gives an overview on the specified UC activities. A UC diagram was utilized to depict the interaction between the user and the SFCS, as well as the relationships between the use cases.

Table 1: Use case descriptions

Use Cases	Use case activities
Performance monitoring	▪ Measure electrical parameters (voltage, current, power) with the smart module ▪ Transmit data in real-time to the ThingsBoard ▪ Store measurement data in the database ▪ Visualize data in the user interface (UI)
Performance analysis	▪ Alarm user when the FC output power is outside the user-defined range ▪ Display alarms on both the Smart Module and the UI
Alarm resolution	▪ Allow the user to resolve alarms through the Smart Module ▪ Communicate alarm resolution to ThingsBoard ▪ Update and display the status in the UI
Energy analysis	▪ Determine the energy produced by the FC over a specified period ▪ Set the time period and display the results via the UI.
Actuator control	▪ Enable remote control of a smart module actuator elements through the UI

Requirements: Based on the environmental analysis and defined UC scenarios, the system requirements for the SFCS were developed. These requirements were further specified and documented during the product development process. The requirements were divided into software, hardware, and smart services. Each requirement was described in a solution-neutral manner using the *SOPHISTEN* wording enhance clarity [25]. Furthermore, the systems requirements were prioritized as mandatory or desirable and were distinguished in functional and non-functional requirements.

Functional breakdown: Taking the overall function of SFCS, its defined use cases and system requirements into account, the main functions were derived and divided into subfunctions in an iterative process until solution patterns or operating principles can be identified for the lowest-level functions. To keep the solution space open, these functions were kept solution-neutral and formulated using noun-verb structures.

System architecture: In this step, the system elements required to realize the desired system functions were identified and derived. It was also determined which functions would be implemented in software, hardware, or both. Design-structure matrices (DSM) and Domain-

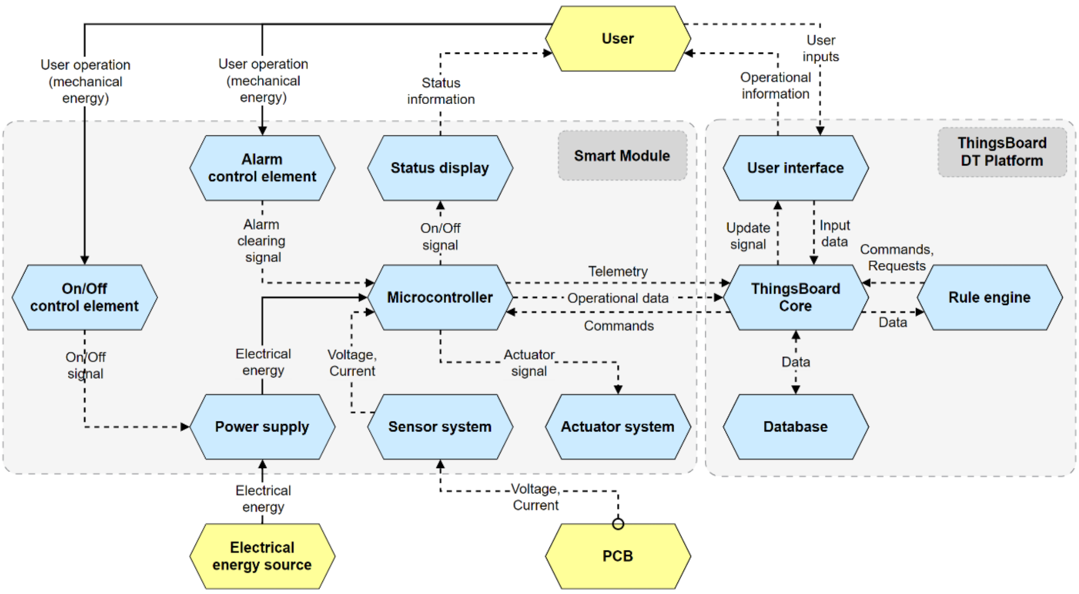


Figure 5: System architecture

mapping matrices (DMM) were used to identify internal system dependencies and eliminate redundant functions and elements, as suggested in literature [20, 26]. The interactions between the derived system elements were modeled in the system architecture, which provides a white box view of the SFCS and shows the interaction between all system elements (see figure 5). As shown, the SFCS is subdivided into the subsystems ‘Smart Module’ and ‘ThingsBoard DT platform’, with the latter representing the logical system architecture of the DT.

System behavior: In this methodological step, the system behavior within the defined use cases is further specified. The system behavior was specified using sequence diagrams, which illustrate the chronological interactions between derived system elements (see figure 6). These diagrams served as the foundation for the subsequent software design.

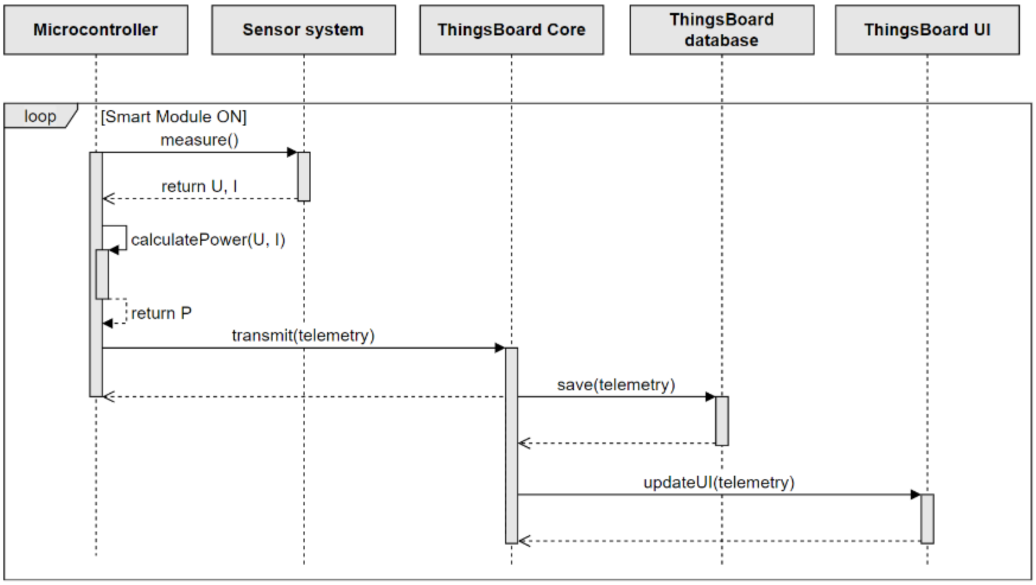


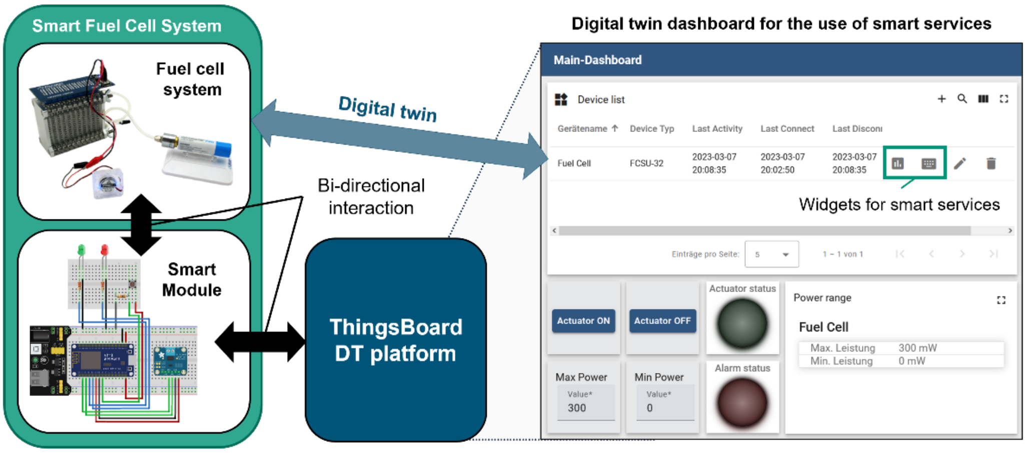
Figure 6: Sequence diagram for the smart service “Performance monitoring”

4.3. Domain specific design and system integration

The authors chose to address the system functions without employing mechanical principles to reduce design complexity. Consequently, the domain-specific design focused on hardware and software design. For hardware design, solution principles for physical components were identified, compared, and selected based on requirements, leading to the SM's assembly. In software design, various implementation approaches were evaluated, with IoT data communication technologies chosen for software functions. The SM and ThingsBoard rule chains were then programmed using the ThingsBoard Arduino SDK to encode the logic for smart services. Component testing verified both hardware and software. Integration tests ensured proper interaction between them, using debugging tools in the programming environments for the SM's microcontroller and ThingsBoard rule chains.

5. Results

The initial development goals for the SFCS have been achieved through the application of the method proposed in this paper. The fuel cell under consideration has been transformed into a Smart Fuel Cell (SFC) through the developed Smart Module. Additionally, a dashboard has been created, allowing users to interact bidirectionally with the SFC and utilize the services. Furthermore, all planned services have been implemented. Iterative approaches were developed to encompass the attributes required for the services and to implement the rule-based logic, based on developed sequence diagrams. Overall, the SFCS can be classified as an DT-SPSS according to the definitions (see figure 7).



Gerätename ↑	Device Typ	Last Activity	Last Connect	Last Disconı
Fuel Cell	FCSU-32	2023-03-07 20:08:35	2023-03-07 20:02:50	2023-03-07 20:08:35

Figure 7: Developed DT-SPSS

6. Discussion

Creating partial models predefined in CONSENS greatly improved system understanding and reduced complexity during the system design phase. For example, modeling the IoT platform within the SFCS clarified interactions between the Smart Module and ThingsBoard. Although ThingsBoard could be viewed as an external element, it was modeled as part of the SoI to better analyze component interactions. This approach was essential for the design process, with sequence diagrams helping to manage complexity and clarify system behavior within the smart services. Nevertheless, the models created during the system design were document based, which prevented them from being interlinked to form a cohesive digital system model – a key advantage of MBSE. This limitation resulted in a higher number of iterations throughout the development process. Another critical consideration is that the CONSENS modeling language, as established, does not adequately address the complexity of the DT-SPSS. Furthermore, the integration of the system model with the digital twin was not explored in detail. The integration of the system model with the digital twin was not thoroughly explored since the DT was defined to have an informational character. For example, using a PaaS IoT platform complicated the inclusion of behavioral models in the digital twin, which could have enabled services such as predictive maintenance and enhanced value creation. However, it is important to note that the stated limitations lay beyond the scope of the formulated research goal.

7. Conclusion and outlook

Within the scope of this paper, the hypothesis that MBSE is suitable for the design of SPSS has been supported. Despite employing a document based MBSE approach, advantages in system understanding and behavioral descriptions were evident. These aspects are particularly significant for systems with high degrees of connectivity, such as SPSS. However, the authors conclude that future research should focus on leveraging computer-aided MBSE approaches for designing DT-SPSS. This is essential for fully realizing the benefits of DT-SPSS, which come from integrating a system model created using MBSE with the digital twin. Such integration would create a single source of truth for all description models throughout the PLC. An additional consideration for future research could be exploring how to systematically integrate the business aspect into the DT-SPSS development process to further enhance value creation.

Acknowledgements

The corresponding author thanks the co-authors for their contribution in assistance during processing of the master's thesis, which formed the basis for this paper.

Literature

[1] ABRAMOVICI, Michael (Hrsg.); STARK, Rainer (Hrsg.): *Smart Product Engineering: Proceedings of the 23rd CIRP Design Conference, Bochum, Germany, March 11th - 13th, 2013*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013

[2] STARK, Rainer ; FRESEMANN, Carina ; LINDOW, Kai: *Development and operation of Digital Twins for technical systems and services*. In: *CIRP Annals* 68 (2019), Nr. 1, S. 129–132

[3] DURÃO, Luiz Fernando C.S. ; ZANCUL, Eduardo ; SCHÜTZER, Klaus: *Digital Twin data architecture for Product-Service Systems*. In: *Procedia CIRP* 121 (2024), S. 79–84

[4] BERTONI, Marco ; BERTONI, Alessandro: *Designing solutions with the product-service systems digital twin* : *What is now and what is next?* In: *Computers in Industry* 138 (2022)

[5] HALSTENBERG, Friedrich: *Methodik zur modellbasierten Systemarchitekturdefinition von Smart-circular Product-Service Systems*. Berlin, Technischen Universität Berlin. Dissertation. 2022

[6] APOSTOLOV, Hristo ; FISCHER, Matthias ; OLIVOTTI, Daniel ; DREYER, Sonja ; BREITNER, Michael H. ; EIGNER, Martin: *Modeling Framework for Integrated, Model-based Development of Product-Service Systems*. In: *Procedia CIRP* 73 (2018), Nr. 4, S. 9–14

[7] QU, Min ; YU, Suihuai ; CHEN, Dengkai ; CHU, Jianjie ; TIAN, Baozhen: *State-of-the-art of design, evaluation, and operation methodologies in product service systems*. In: *Computers in Industry* 77 (2016), S. 1–14

[8] CHOWDHURY, Soumitra ; HAFTOR, Darek ; PASHKEVICH, Natallia: *Smart Product-Service Systems (Smart PSS) in Industrial Firms: A Literature Review*. In: *Procedia CIRP* 73 (2018), S. 26–31

[9] CHATTI, Sami ; LAPERRIÈRE, Luc ; REINHART, Gunther ; TOLIO, Tullio: *CIRP encyclopedia of production engineering*. New York, NY : Springer Berlin Heidelberg, 2019

[10] FREITAG, Mike ; WIESNER, Stefan: *Smart Service Lifecycle Management*. In: *Industrie 4.0 Management* 2019 (2019), Nr. 5, S. 35–39

[11] WILKING, Fabian ; SCHLEICH, Benjamin ; WARTZACK, Sandro: *DIGITAL TWINS - DEFINITIONS, CLASSES AND BUSINESS SCENARIOS FOR DIFFERENT INDUSTRY SECTORS*. In: *Proceedings of the Design Society* 1 (2021), S. 1293–1302

[12] MADNI, Azad ; MADNI, Carla ; LUCERO, Scott: *Leveraging Digital Twin Technology in Model-Based Systems Engineering*. In: *Systems* 7 (2019), Nr. 1, S. 7

[13] BRISSAUD, Daniel ; SAKAO, Tomohiko ; RIEL, Andreas ; ERKOYUNCU, John Ahmet: *Designing value-driven solutions: The evolution of industrial product-service systems*. In: *CIRP Annals* 71 (2022), Nr. 2, S. 553–575

[14] LINDOW, Kai: *Horizontal Value Creation through IoT and Digital Twin Data*. In: *The Proceedings of Mechanical Engineering Congress, Japan* 2022 (2022), Nr. 0, C121-01

[15] FETT, Michel ; WILKING, Fabian ; GOETZ, Stefan ; KIRCHNER, Eckhard ; WARTZACK, Sandro: *A Literature Review on the Development and Creation of Digital Twins, Cyber-Physical Systems, and Product-Service Systems*. In: *Sensors (Basel, Switzerland)* 23 (2023), Nr. 24

[16] ZHENG, Pai ; WANG, Zuoxu ; CHEN, Chun-Hsien ; PHENG KHOO, Li: *A survey of smart product-service systems: Key aspects, challenges and future perspectives*. In: *Advanced Engineering Informatics* 42 (2019), S. 100973

[17] GOGINENI, Sonika Krishna ; RIEDELSHEIMER, Theresa ; STARK, Rainer: *Systematic product development methodology for customizable IoT devices*. In: *Procedia CIRP* 84 (2019), S. 393–399

[18] [18] RIEDELSHEIMER, Theresa ; GOGINENI, Sonika ; STARK, Rainer: *Methodology to develop Digital Twins for energy efficient customizable IoT-Products*. In: *Procedia CIRP* 98 (2021), S. 258–263

[19] KAISER, Lydia: *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*. Paderborn, Universität Paderborn, Fakultät Maschinenbau. Dissertation. 2013

[20] HABERFELLNER, Reinhard ; WECK, Olivier de ; FRICKE, Ernst ; VÖSSNER, Siegfried: *Systems Engineering*. Cham : Springer International Publishing, 2019

[21] MADNI, Azad M. (Hrsg.); AUGUSTINE, Norman (Hrsg.); SIEVERS, Michael (Hrsg.): *Handbook of Model-Based Systems Engineering*. Cham : Springer International Publishing, 2023

[22] ESTEFAN, Jeff: *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. In: *INCOSE MBSE Initiative* (2008)

[23] MORKEVICIUS, Aurelijus ; ALEKSANDRAVICIENE, Aiste ; MAZEIKA, Donatas ; BISIKIRSKIENE, Lina ; STROLIA, Zilvinas: *MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems*. In: *INCOSE International Symposium* 27 (2017), Nr. 1, S. 136–150

[24] ESTEFAN, Jeff A. ; WEILKIENS, Tim: MBSE Methodologies. In: MADNI, Azad M.; AUGUSTINE, Norman; SIEVERS, Michael (Hrsg.): *Handbook of Model-Based Systems Engineering*. Cham : Springer International Publishing, 2023, S. 47–85

[25] DIE SOPHISTEN: *MASTER* : *Schablonen für alle Fälle*. Nürnberg, 2019

[26] DUMITRESCU, Roman (Hrsg.); FLEUTER, Markus (Hrsg.): *Intelligenter Separator* : *Optimale Veredelung von Lebensmitteln*. 1. Auflage 2019. Berlin : Springer Berlin; Springer Vieweg, 2019 (Intelligente Technische Systeme – Lösungen aus dem Spitzencluster it's OWL)