AMCIS 2022 Proceedings

SIG SAND - Systems Analysis and Design

Aug 10th, 12:00 AM

Closing the Implementation Gap of Digital Twins

Hendrik Wache
Chemnitz University of Technology, hendrik.wache@wirtschaft.tu-chemnitz.de

Eric Jan Hübner
Chemnitz University of Technology, jan-eric.huebner@wiwi.tu-chemnitz.de

Sarah Hönigsberg
Chemnitz University of Technology, sarah.hoenigsberg@wirtschaft.tu-chemnitz.de

Barbara Dinter
Chemnitz University of Technology, barbara.dinter@wirtschaft.tu-chemnitz.de

CLOSING THE IMPLEMENTATION GAP OF DIGITAL TWINS

Completed Research

Hendrik Wache
Chemnitz University of Technology
hendrik.wache
@wirtschaft.tu-chemnitz.de

Jan Eric Hübner
Chemnitz University of Technology
jan-eric.huebner
@wirtschaft.tu-chemnitz.de

Sarah Hönigsberg
Chemnitz University of Technology
sarah.hoenigsberg
@wirtschaft.tu-chemnitz.de

Barbara Dinter
Chemnitz University of Technology
barbara.dinter
@wirtschaft.tu-chemnitz.de

Abstract

Since the manufacturing industry is facing increasingly advancing digitalization, digital twins (DT) have become a popular means for integrating various actors' value creation using a smart product. DTs are information systems that connect the physical and virtual worlds. The design of DTs is time-consuming, expensive, and lacks appropriate prescriptive design knowledge for its development. Design principles (DP) represent a mechanism to codify design knowledge into prescriptive knowledge. However, the mostly abstract DPs are often difficult for practitioners to operationalize during software development projects, rendering the design knowledge difficult to access. The paper at hand addresses these issues by providing a reference model for DT development as a semi-abstract artifact. The model has been constructed by drawing on a literature review and empirical cases in the manufacturing industry. The reference model includes multiple adaptation mechanisms to ensure a flexible development of company-specific DTs.

Keywords

Reference Model, Digital Twin, Information System Modeling, Design Principles.

Introduction

Today's world is continually shaped by advancing digitalization. Digitalization has a particular impact on the manufacturing industry, to the extent that it is referred to as the fourth industrial revolution (industry 4.0, hereafter I4.0). It involves using advanced information technologies that connect industry and its plants with the internet of things and services (Kagermann et al. 2013). I4.0 enables an increasing convergence between the physical and the virtual space. Digital twins (DT) embody this convergence by connecting physical assets (e.g., a production plant) and virtual components (e.g., a behavioral model of the plant) in particular. DTs have a large amount of data, such as product specifications, production process models, operational performance data, or even written knowledge. However, the numerous virtual models that perform analyses and simulations, thereby generating descriptive, predictive, and prescriptive knowledge, are the focus of DTs. DTs can be considered to be oriented towards collaborative value creation in a network of actors surrounding a smart object/product, which enables the integration of (knowledge) resources for mutual benefits (Beverungen et al. 2019).

A DT enables companies to achieve considerable operational added value and can become the central point of contact for all data in the relevant plant by integrating it with existing operational systems such as enterprise resource planning, customer relationship management, or product data management. However, the added value of DTs is also accompanied by challenges. For example, high development costs and the high maintenance effort make widespread implementation in organizations difficult. Developing a DT from scratch is a time-consuming challenge, especially for smaller companies typical for the manufacturing industry. This challenge is complicated by company-specific requirements and a high level of development complexity, often leading to development without sufficient conceptual planning. Against this background, there is a particular lack of design-oriented studies for supporting the development of I4.0 systems in general (Baiyere et al. 2020) and DTs, especially with guidance for developers in practice.

Although previous research described some aspects of DT development, there is a lack of synthesis of findings to support development efforts. On the one hand, practical DT developments are taking place in the industry. However, DTs are often considered too technical. Their integration into the business processes and the interaction between DTs and the human workforce is not sufficiently considered, often leading to merely prototypical implementations. On the other hand, design knowledge is emerging from the scientific discourse on DTs. In the information systems (IS) discipline, this design knowledge is often captured in so-called design principles (DP), which are an abstracted representation of the characteristics of an artifact condensed into a prescriptive statement to be considered for the design. In recent years there has been increasing discussion about the extent to which these DPs are practical (Chandra Kruse et al. 2016; Schoormann et al. 2021), i.e., to what extent a system developer who wants to develop a DT, for example, is properly supported by this abstract knowledge. Thus, a two-sided problem arises: In practice, there is a lack of best practices to design DTs as an integral part of corporate value creation, and from the scientific side, there is a lack of a knowledge transfer mechanism that makes the academic findings on DTs easy to implement for practitioners.

In a multi-year design study on the development of a DT platform, we have shown that the developed DPs were useful and helpful for knowledge transfer in the academic environment (Wache and Dinter 2021). However, an "implementation gap" arose in using the DPs for practitioners in the project. This gap arose because the DPs could not be operationalized by the various developers and were perceived purely as "food for thought" during the implementation. Therefore, it is necessary to find a less abstract vehicle for the design knowledge, which should be able to integrate the DPs, is universally valid, and yet is still close to implementation to enable the translation of the knowledge by practitioners to the instance level. Therefore, this work presents an artifact that compiles the best practices behind the development of DTs in such a way that an application of the design knowledge can be made in different contexts by different developing companies. For this purpose, reference modeling is suitable, as it simplifies the design process and the actual development by bringing together best practices, recommendations, and knowledge from a research field and making them usable (Becker et al. 2007). This paper synthesizes acquired knowledge from prior DPs, a literature review, focus group interviews, and workshops to create a reusable conceptual artifact that contains prescriptive knowledge about DT design and development. This leads to the following research question: How should a DT reference model be built to make design knowledge available to IS scholars and practitioners and close the implementation gap?

The remainder of the paper is organized as follows: After we described the conceptual foundations of our research, the research design is outlined. We develop the reference model in the subsequent section, followed by an evaluation. The paper concludes after a discussion of the results.

Conceptual Foundations

Digital Twins

The first notion of using a DT originated in product lifecycle management, which conceived this twin as a virtual representation of a physical system. The DT is supposed to contain all data about and current status data of its physical counterpart (Grieves 2014; Grieves and Vickers 2017). However, the concept of the DT is now also used outside the field of product lifecycle management, for instance, in the manufacturing sector (Grieves 2014; Shafto et al. 2012). According to Rosen et al. (2015), manufacturing systems will have to operate more autonomously in the future to meet the emerging challenges. DTs are expected to help overcome these challenges by bridging the gap between the real and the digital world and providing comprehensive data across all stages (Glaessgen and Stargel 2012; Rosen et al. 2015). In this sense, DTs can be understood either as enablers or subtypes of cyber-physical systems (Dietz and Pernul 2020). In general, however, DTs are not just a collection of different digital artifacts but form a system in which these artifacts are connected in a structured way and consequently contain meta-information (Rosen et al. 2015). Several components need to be considered: a physical component, a virtual component, a service system, the data, and a connection between all of them (Tao et al. 2019). The use cases for DTs are diverse, spanning several dimensions from product design and production to optimization and maintenance (Rosen et al. 2015; Tao et al. 2019), but are commonly focused on control, simulation, and monitoring, with data becoming a central driver. Tao et al. (2019) discuss the use of DTs in value chains in which the DT provides services to the customers. Therefore, the DT can be considered a platform to enable human engagement in I4.0 and to support the value creation of multiple actors.

Abstract Design Knowledge and Reference Models

DPs represent prescriptive knowledge intended to aid practitioners in translating abstract concepts into a more tangible form for practical application. However, this is often not sufficient because the development of IT artifacts is a continuous, complex, and context-specific process (Chandra Kruse et al. 2016; Sein et al. 2011). Chandra Kruse et al. (2016) and Amabile et al. (1996) observe that the successful application of DPs depends on the designers' knowledge base, resulting in something that could be described as an implementation gap. This gap describes a distance between the proposed solution and available resources, such as design knowledge, ranging from tacit knowledge among developers to design principles and technical models. In a sense, the issue also relates to a problem of bridging between two experts, one explicating his knowledge and the other applying it. To narrow this implementation gap, this work attempts to propose a reference model for DT platforms, as designers commonly use such models as a basis for their work during the design phase of IS. Reference models constitute a simplified or optimized representation of a system, which enables the derivation of design proposals from an ideal concept (Rosemann and van der Aalst 2007). Furthermore, our reference model follows the understanding of Becker et al. (2007) and Rosemann and van der Aalst (2007) and is intended to apply to a class of abstract application areas and thus exhibit both generality and adaptability. Such a generally applicable reference model has to be transformed into a specific model for practical application in most cases (Fettke and Loos 2003). Therefore, we develop specific mechanisms that allow the reference model to be customized.

Research Approach

We aim to develop a reference model for manufacturing DTs by synthesizing the design outcomes of four case companies. We aim to support companies in their I4.0 initiatives by providing best practices for DT design in an easily applicable way. For this purpose, the design science research framework of Kuechler and Vaishnavi (2008) is combined with reference modeling approaches (Rosemann and van der Aalst 2007).

Awareness of the problem

In 2018, we launched a multi-year design study on DT development in the manufacturing context. Four mechanical engineering companies participated, one developing a DT in sales, two a DT in product lifecycle management, and one a DT in service and maintenance. The data collection on which the DPs and the continuing reference model are based was multifaceted: qualitative questionnaires, eight hours of semi-structured interviews (Fontana and Frey 1994), six hours of focus group discussions (Morgan 1997), 20 hours of workshops, and more than 30 hours of direct and indirect observations (Mayring 2004). The data were both open and deductively coded. The design knowledge was translated into DPs following Gregor, Chandra Kruse, & Seidel (2020). The DT developers were provided with these DPs but struggled to operationalize them. Overall, we concluded that the problem was an implementation gap that made it difficult for practitioners to incorporate the complex design knowledge of DTs into the development process.

Suggestion

Our suggestion is to develop an adaptive reference model on a conceptual level, which means it should abstract the used technology. This way, it is general enough to represent the design knowledge in a case-independent way but at the same time gives enough support to develop complex systems like the DT. This conceptual level allows heterogenic groups of DT developers, target users, and researchers to discuss and plan DT implementations tailored to the companies' value creation. The envisioned reference model closes the implementation gap and thereby addresses the scientific side of the problem of DT development. While there are already some reference models for DT, they neither address codifying design knowledge nor target software developers (Bevilacqua et al. 2020; Zheng and Sivabalan 2020). We refer to an established approach to derive an adaptive reference model in a design study based on literature and empirical input (Hönigsberg et al. 2019). Our goal is to reduce the development effort for DTs and imprint the gained design knowledge (DPs) into a reference model, which can be best realized by combining a configurative and a generic adaptive reference model. Hence, we combine two types of adaptive reference models. Thus, with the suggested solution, an instance-specific model can be configured and then manually adapted for an optimal fit.

Development

The basic knowledge for developing DTs stems from our multi-year study on DT design. In the development process, we captured theory ingrained design knowledge in DPs to guide the DT design in our four manufacturing companies. These DPs were refined in the empirical cases. When the project team encountered the "implementation gap" problem, the abstract DPs were reflected and detailed, drawing on DT literature (Wache and Dinter 2021). In the literature review, the Work System Method (Alter 2013) was used as a framework for structuring the scientific discourse of DTs, to understand DTs as a socio-technical system and thereby addressing the practitioner side of the problem of DT development, the over-technical DT understanding. A multi-view reference model was derived from the literature review results, explicating the prescriptive knowledge of the DP from the research case. The intermediate results were validated during the development in a focus group and workshops with 6-15 participants from our case companies. First, we validated the literature-based assumptions on which DT aspects are relevant for specific implementation cases in focus group interviews (cf. choice board and configuration rules). Second, we validated the literature-derived DT functions and corresponding architectural assumptions in workshops (cf. reference model views). The functions with corresponding architectural components were refined and logically arranged to derive use case and architectural diagrams.

Evaluation

The reference model was applied to the four scenarios of the case companies to evaluate the applicability and adequacy of the results. This scenario-based evaluation was discussed in expert interviews with four developers of DT platforms involved in the DT development projects, and two IS scholars who specialized in design research and DP development. The interviews lasted approximately 30 minutes each and included a presentation and demonstration of the reference model followed by a semi-structured interview using an interview guide. The results of the interviews were analyzed to identify potential improvements and gather additional input. The revised reference model was presented to the interviewees to confirm that the improvements were satisfactory. After one iteration, the results were deemed adequate.

The Digital Twin Project - Construction of the Reference Model

Awareness of the Implementation Gap and a Suggestion to Close it

During the design study's intensive requirements analysis and problem formulation from 2018 to the beginning of 2020, interim results were repeatedly discussed and evaluated with the participating companies. However, it became increasingly apparent that both the software development companies and the mechanical engineering companies found the rather theoretical results of the design study difficult to access. On the one hand, there was an impression of "cryptic research babble" and an attitude of "that's nice for research, but I don't need all that in practice." On the other hand, the more detailed DT design knowledge available was overly technical and inadequate to discuss the DT design as an integral part of the case companies' value creation. The companies described an implementation gap in which the distance from the abstracted knowledge to their system development reality is too big, and the overly technical DT descriptions are too far from the companies' scenarios.

To close this implementation gap, the design project results were transferred into a semi-abstract design artifact. Halfway between the discussed instance design and the abstract DPs, conceptual models for design description needed to be developed. In our cases, a dilemma arose as, on one hand, the users of the design knowledge preferred a representation as close as possible to the implementation, so little transfer efforts and few adaptations to the own implementation are necessary. On the other hand, this representation made the results less generalizable, whereby the design knowledge exhibited a small projectability to the other case companies (Vom Brocke et al. 2020). This dilemma is also known in other design knowledge capture and transfer approaches, namely in reference modeling (Becker et al. 2007). Therefore, our proposal to solve the implementation gap problem described above is to apply the solution mechanism of reference modeling to find a middle ground between abstract and detailed guidance. More specifically, adaptive reference modeling was applied to address our described dilemma. Here, the design knowledge from the research cases is transferred into a model that gives generally valid implementation suggestions that are adaptable to the instance case. The specific design of actors and services can be configured in an implementation-specific way via an adaptation mechanism in the reference model (Rosemann and van der Aalst 2007).

Development of the Reference Model

The DT literature was analyzed using the Work System Method as a lens, and the resulting aspects of DTs could be classified into three main topics (Wache and Dinter 2020). First, with Strategy & Environment, Processes & Activity, Actor and Product Level, the type of DT and its deployment is described. The type and the use determine the DT's functionality and structure. Second, Information & Analysis and Service & Function describe what functions it should provide during use. Third, Technological Link and Infrastructure describe the underlying technological basis of a DT Platform. The literature review results lead to the following three components of the reference model: the choice board, the modeling views (functional and architectural view), and the configuration rules.

Choice Board: The following table represents the choice board of the reference model. On the left side of Table 1, a question is stated, and on the right side, there is a choice to answer this question. The questions can be answered from top to bottom, and at the end of this configuration, the reference model suggests a functional and architectural scope for the desired DT. This tool can be used to capture the vision for the DT to be developed at an early planning stage. This type of morphological choice board is well suited to represent the variable solution space for reference models. The grey highlighting exemplifies one of our case scenarios as a configuration in the reference model (grey = selected, white = not selected). The configuration result is highlighted in the functional and architectural view as well.

Question	**Choice**			
Which corporate objective will the DT support?	Product lifecycle management	Product-service system	Sustainability management	Sales management
In which life cycle phase will the DT be used?	Design/plan	Production/build	Service/run	
Who will use the DT?	Customer	Producer /internal	Supplier spanning	
To which product level will the DT refer?	Unit level/component	System level/machine/plant	System of systems /production line/factory	

Table 1. Choice Board

Functional View and Architectural View: The views have been modeled using the Unified Modeling Language (UML) due to its widespread use. The choice of a widely used standard for system modeling supports our intention to close the implementation gap of DTs by choosing an easily accessible and understandable medium for developers, target users of the DT, and researchers alike. In addition to input from the literature, three DPs from the multi-year study were used to inform the reference model's construction. The DPs in the short form are (1) Cyber-physical (re-) configurability, (2) Smartness of the product, and (3) IT platform with a microservice architecture as a boundary object (Wache and Dinter 2021). The functional view (Figure 1) was modeled as a use case diagram. Thus, not only the different functions and activities but also the associated actors could be represented. Based on the literature on DTs, the functions were grouped into three phases: Configuration (Plan), Order & Production (Build), and Operations (Run). This grouping reflects that DT can be used in the planning phase, in the production process, or in the operation of the finished plant. In addition, some administration functions have been identified that need to be supported regardless of the focused phase. Corresponding to this functional view, an architectural view was derived from the literature and our cases. For clarity, a three-layer architecture was chosen (Figure 2). Front ends were modeled for the various actors in the graphical user interface (GUI) layer; in the application view, components behind a central access component represent subsystems that are logically linked to certain functional scopes of the functional view. The different components, in turn, access services, which provide partial functionalities. For example, the configuration component accesses the physical and virtual configuration service to configure the machine as well as the associated sensors and data flows. The data models to be considered are represented in the persistence layer and possible interface systems in the external system view.

The first DP *Cyber-physical (re-) configurability* is exemplified by the presence of a configuration function in the use case diagram as well as the presence of the configuration component with both physical and virtual configuration services. The second DP *Smartness of the product* is embodied by the fact that in the use case diagram, analysis functions such as simulation and optimization are made available to the user in all lifecycle phases of the DT. This DP is reflected by the analysis service's central position, which is interconnected with almost all components in the architecture view. The third DP *IT platform with a microservice architecture as a boundary object* leads to the system being designed as a common platform from the customer to the supplier. This DP is reflected both in the use case diagram by the actors and in the architecture diagram by the various front ends and the possibility of integrating external systems.

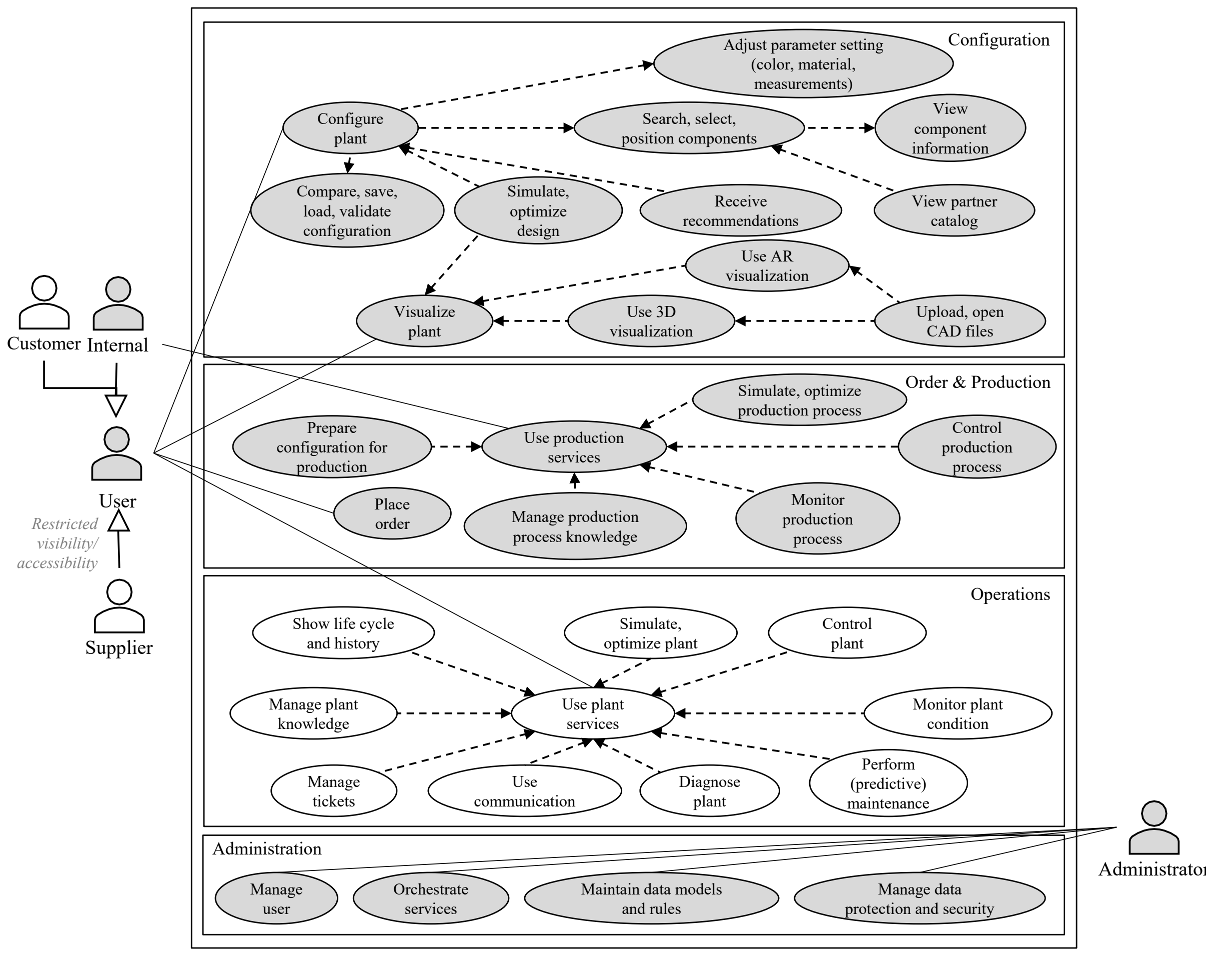

Figure 1 Functional View

Adaptation Mechanisms and Rules: The type and use of DTs determine the system's functionality and architecture. Thus, there is a logical relationship between the choices in the choice board and the individual elements in the functional and architectural view (cf. highlighting in Table 1, Figures 1 and 2). The DT can be configured on a conceptual level using the choice board. Several rules have been defined that select functions and the corresponding architecture components when a choice is made in the choice board. The configured model is created in the first step, which specifies the recommended scope for the two diagrams for a specific DT to be developed. In the second step, the model can be adapted (manually) with a generic adaptation to generate a better fit for the specific application scenario. An example of a combination of several such rules affecting both views: IF Design/Plan AS life cycle phase THEN INCLUDE Configure plant AS Function AND Configuration component AS Architectural Component USING Physical configuration service AND Virtual configuration Service. This chaining of rules does not correspond to all rules triggered when Design/Plan is selected in the Choice Board but corresponds to a continuous example from the Choice Board to the architecture view. For example, a rule for generic adaptation is: IF REMOVE Place order AS Function THEN REMOVE Order Service AS Architectural component AND Order Model AS Persistence Model. This type of rule does not configure the model in the classical sense but ensures consistency between views when manipulated manually.

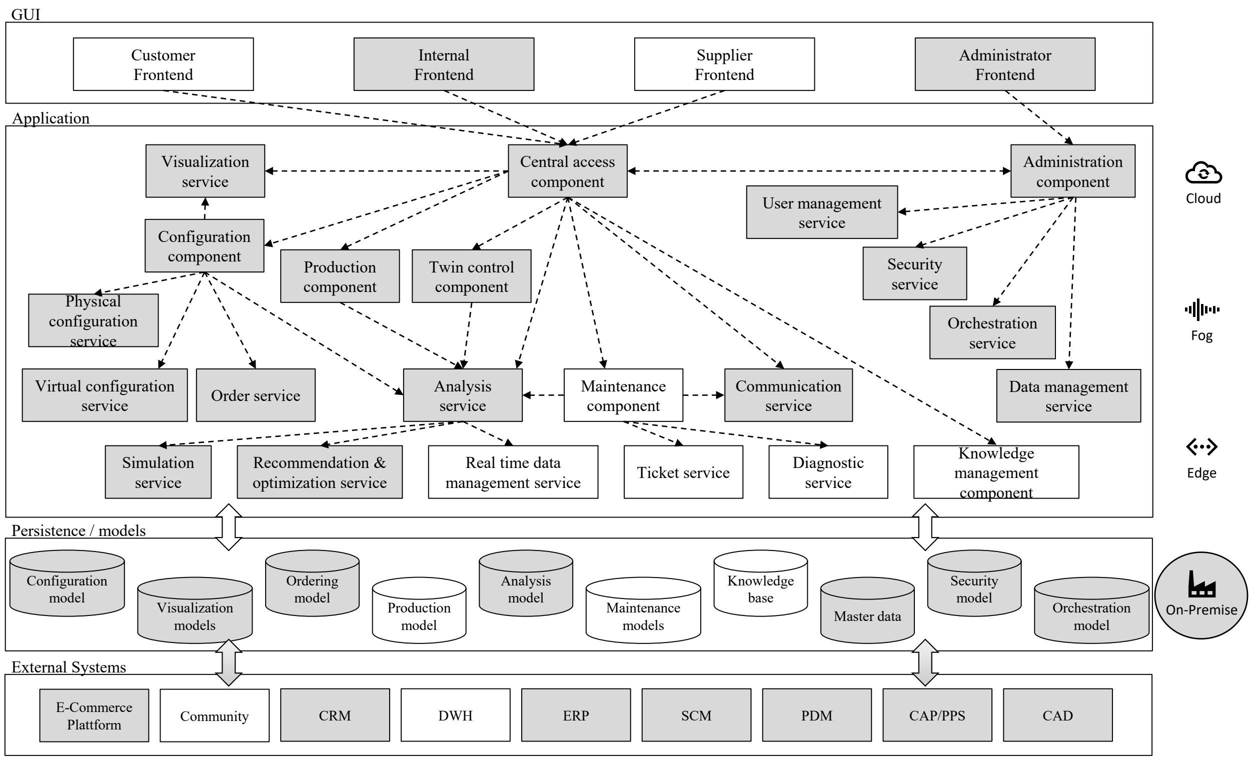

Figure 2 Architectural View

Evaluation of the Reference Model

Several interviews were conducted for the evaluation, involving four DT developers and two researchers, one focusing on DPs, and another with a background in reference models. The DPs and the reference model were shown and explained to the interviewees before they answered some questions of the interview guide. A summary of the collected feedback is provided hereafter. Applying the reference model to the case companies' scenarios was deemed suitable support by the developers, above all for an early development phase. All interviewees were positive about whether the reference model is helpful for developers of DT platforms. It was confirmed by the four developers that the reference model makes the rather abstract DPs more accessible for the target group of practitioners and that they would prefer the reference model for their development projects. However, the extent of support provided by the reference model would depend on the application context and the individuals. Thus, the interviews with developers revealed a different use of DPs and reference model depending, e.g., on the assumed software development methodology (waterfall model, agile software development). There was unanimity among all interviewees that the DPs play a strategic role in system development, whereas the reference model is used during operational software development. Concerning DPs, there were differing views among developers and researchers as to whether DPs should describe a system in full (practitioner viewpoint) or rather have a focus on a thematic area (academic viewpoint). With regard to the reference model, both interviewed groups emphasized that its extensibility is of great importance so that further DPs could be reflected and the reference model applies to different contexts, thus facilitating the instantiation of the design knowledge. With respect to extensibility, both groups repeatedly expressed the desire for the addition of a data view that maps the DT data schemas. In addition, the interviewees in both groups were missing an overarching process perspective or meta-process that outlines how the target group of developers should use the reference model to incorporate and implement the resulting suggestions into the development process.

After the interviews, the revision of the reference model focused on extending the views to homogenize their granularity. Thus, the production part of the use case and the architecture diagram were extended by elements.

Discussion

Our study developed a reference model to address the implementation gap by providing designers/practitioners with a configurative tool that integrates the abstract DPs into more practical, use-case-based approaches for DT design. We successfully answered the research question on how a DT reference model should be constituted to make design knowledge available to IS scholars and practitioners to close the implementation gap. The addition of our reference model converts the required knowledge base for designers by integrating DPs into a guiding architecture while still allowing practitioners the necessary freedom to develop a DT based on their needs. This principle is generally useful when technically independent conceptual models are used in development projects to synchronize the world view of the various project stakeholders. We assume that our model can also be used for companies in the same industry. Using a socio-technical lens to extract the characteristics of DTs from the literature and transfer the findings into a reference model leveraging DPs from a DT design study, we were able to address both sides of the problem designers are facing while developing DTs. Our study addresses a gap in the research on the design of I4.0 technologies such as DTs. In addition to accumulating knowledge about systems like DTs (Baiyere et al. 2020), we also focus on the mechanism of making this knowledge accessible to practitioners. This is an important undertaking because companies need the necessary technical expertise to successfully implement digitalization initiatives in their companies (Legner et al. 2017; Vial 2019).

Of course, not all DPs are 'cryptic research babble'; there is a continuum between abstraction and specificity in which DPs exist (Wache et al. 2022). Nonetheless, it can be observed that nuances of the encoded design knowledge in DPs are lost in use. The transfer of DPs into a form closer to and more understandable for the target group of developers was feasible. This transfer was tested in an evaluation, and it was shown that the resulting format is more accessible to practitioners. Thus, it could be demonstrated that the implementation gap could be successfully addressed. With this transfer of prescriptive knowledge to the target group of practitioners, our study addresses one of the goals of design-oriented IS research (Chandra Kruse et al. 2016; Gregor et al. 2020). Design-oriented IS research seeks to accumulate instance knowledge and make it accessible to other contexts and scenarios. In this context, design knowledge can be characterized as descriptive and prescriptive. Other recent work is concerned with transferring descriptive knowledge into prescriptive knowledge (Möller et al. 2021), which benefits the accumulation of prescriptive knowledge. Our work can also be located in the process of knowledge generation and transfer but starts later, as we de-abstract prescriptive design knowledge, i.e., express it less abstractly and more contextually to make it more accessible to practitioners. Our work ties in with the findings of Möller et al. (2021) by representing different aspects/characteristics/degrees of detail of the design knowledge spectrum through our proposed reference model. Furthermore, the developed reference models represent a continuation of DTs' conceptualization (van der Valk et al. 2021), where the development of DPs and reference models for DTs are necessary next research steps. Van der Valk et al. (2021) argue that practitioners can improve their understanding of DTs by critically examining their proposed archetypes. Our work contributes another building block to this more accessible knowledge base on DTs for practitioners, introducing a DP-based reference model. It enables practitioners to develop a company-specific DT, which acts as a guideline for developing DTs.

Nevertheless, our work has some limitations. Our research examined DTs through a socio-technical lens, considering DTs as a platform that enables human engagement in an I4.0 context, thereby limiting the applicability of our reference model to the domain of manufacturing. Our reference model is not readily applicable to other DT domains such as buildings or healthcare. The practitioners evaluated the content of our proposed reference model based on their experience of developing a DT platform during our multi-year study on DTs. The two researchers, one from the field of DPs and one from the field of reference models, had no connection to our research case and thus provided an external perspective onto the reference model.

We acknowledge that further evaluation is required to validate our findings beyond the scope of our research case by transferring the reference model onto other cases. During our evaluation of the reference model, it became apparent that the reference model could be extended by several additional and zoomed-in detail views to increase its overall usefulness. Thus, the current scope of the reference model can be seen as a limitation.

Conclusion

Our study aims to create a reference model for DTs to support developers during the development process. We address an implementation gap perceived by practitioners when they are confronted with abstract DPs and convert them into implementation. Based on the characteristics of DTs from the literature and our case companies, a multi-view adaptive reference model has been developed. As shown in the evaluation, the reference model represents an abstract knowledge vehicle that can support the DT development already in the early phases of development projects. In summary, our research contributes to the knowledge base by integrating reference modeling with the more abstract DPs to create a semi-abstract knowledge artifact. Future research should extend the evaluation to more cases, and further views and view refinements for the DT reference model might be developed. A particular focus should be on further closing the implementation gap by expanding the reference model from the abstract level to a level that is no longer independent of technology and contains specific technical implementation approaches.

Acknowledgements

The research in this paper was supported by a grant from the German Ministry for Research and Education (BMBF), project name: Co-TWIN, no: 02P17D146.

REFERENCES

Alter, S. 2013. “Work System Theory: Overview of Core Concept, Extensions, and Challenges for the Future,” Journal of the Association for Information Systems (14:2), pp. 72–121.

Amabile, T. M., Collins, M. A., Conti, R., Phillips, E., Picariello, M., Ruscio, J., and Whitney, D. 1996. Creativity in Context: Update to the Social Psychology of Creativity, New York: Routledge.

Baiyere, A., Topi, H., Venkatesh, V., Wyatt, J., and Donnellan, B. 2020. “Internet of Things (IoT) - A Research Agenda for Information Systems,” Communications of the Association for Information Systems (47:1), pp. 564–589.

Becker, J., Delfmann, P., and Knackstedt, R. 2007. “Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models,” in Reference Modeling Efficient Information Systems Design Through Reuse of Information Models, J. Becker and P. Delfmann (eds.), Heidelberg: Physica, pp. 27–58.

Beverungen, D., Müller, O., Matzner, M., Mendling, J., and vom Brocke, J. 2019. “Conceptualizing Smart Service Systems,” Electronic Markets (29:1), pp. 7–18.

Bevilacqua, M., Bottani, E., Ciarapica, F. E., Costantino, F., Donato, L. Di, Ferraro, A., Mazzuto, G., Monteriù, A., Nardini, G., Ortenzi, M., Paroncini, M., Pirozzi, M., Prist, M., Quatrini, E., Tronci, M., and Vignali, G. 2020. “Digital Twin Reference Model Development to Prevent Operators’ Risk in Process Plants,” Sustainability (12:3), pp. 1–17.

Vom Brocke, J., Winter, R., Hevner, A., and Maedche, A. 2020. “Special Issue Editorial – Accumulation and Evolution of Design Knowledge in Design Science Research: A Journey through Time and Space,” Journal of the Association for Information Systems (21:3), pp. 520–544.

Chandra Kruse, L., Seidel, S., and Purao, S. 2016. “Making Use of Design Principles,” in DESRIST 2016: Tackling Society’s Grand Challenges with Design Science, J. Parsons, T. Tuunanen, J. Venable, B. Donnellan, M. Helfert, and J. Kenneally (eds.), St. John’s: Springer, pp. 37–51.

Dietz, M., and Pernul, G. 2020. “Digital Twin: Empowering Enterprises Towards a System-of-Systems Approach,” Business and Information Systems Engineering (62:2), pp. 179–184.

Fettke, P., and Loos, P. 2003. “Multiperspective Evaluation of Reference Models – Towards a Framework,” in Conceptual Modeling for Novel Application Domains: ER 2003, Lecture Notes in Computer Science (LNCS) (Vol. 2814), M. A. Jeusfeld and Ó. Pastor (eds.), Berlin: Springer, pp. 80–91.

Fontana, A., and Frey, J. H. 1994. “Interviewing: The Arts of Science,” in Handbook of Qualitative Research, N. K. Denzin and Y. S. Lincoln (eds.), Thousand Oaks: SAGE Publications, Inc., pp. 361–376.

Glaessgen, E., and Stargel, D. 2012. “The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles,” in 53rd Structures, Structural Dynamics and Materials Conference, Honolulu, pp. 1–14.

Gregor, S., Chandra Kruse, L., and Seidel, S. 2020. “The Anatomy of a Design Principle,” Journal of the Association for Information Systems (21:6), pp. 1622–1652.

Grieves, M. 2014. “Digital Twin: Manufacturing Excellence through Virtual Factory Replication,” Digital Twin White Paper.

Grieves, M., and Vickers, J. 2017. “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems,” in Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, F.-J. Kahlen, S. Flumerfelt, and A. Alves (eds.), Cham: Springer, pp. 85–113.

Hönigsberg, S., Kollwitz, C., and Dinter, B. 2019. “Designing a Reference Model for Digital Product Configurators,” in 14th International Conference on Wirtschaftsinformatik, Siegen, pp. 214–228.

Kagermann, H., Wahlster, W., and Helbig, J. 2013. “Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Final Report of the Industrie 4.0 Working Group,” Frankfurt/Main.

Kuechler, B., and Vaishnavi, V. 2008. “On Theory Development in Design Science Research: Anatomy of a Research Project,” European Journal of Information Systems (17:5), pp. 489–504.

Legner, C., Eymann, T., Hess, T., Matt, C., Böhmann, T., Drews, P., Mädche, A., Urbach, N., and Ahlemann, F. 2017. “Digitalization: Opportunity and Challenge for the Business and Information Systems Engineering Community,” Business and Information Systems Engineering (59:4), pp. 301–308.

Mayring, P. 2004. “Qualitative Content Analysis,” in A Companion to Qualitative Research, U. Flick, E. von Kardoff, and I. Steinke (eds.), London: SAGE Publications, Inc., pp. 266–270.

Möller, F., Haße, H., Azkan, C., van der Valk, H., and Otto, B. 2021. “Design of Goal-Oriented Artifacts from Morphological Taxonomies: Progression from Descriptive to Prescriptive Design Knowledge,” in 16th International Conference on Wirtschaftsinformatik, Duisburg-Essen, pp. 1–17.

Morgan, D. L. 1997. The Focus Group Guidebook, Thousand Oaks: SAGE Publications, Inc.

Rosemann, M., and van der Aalst, W. M. P. 2007. “A Configurable Reference Modelling Language,” Information Systems (32:1), pp. 1–23.

Rosen, R., Von Wichert, G., Lo, G., and Bettenhausen, K. D. 2015. “About the Importance of Autonomy and Digital Twins for the Future of Manufacturing,” IFAC-PapersOnLine (28:3), pp. 567–572.

Schoormann, T., Möller, F., and Hansen, M. R. P. 2021. “How Do Researchers (Re-)Use Design Principles: An Inductive Analysis of Cumulative Research,” in DESRIST 2021: The Next Wave of Sociotechnical Design, L. Chandra Kruse, S. Seidel, and G. I. Hausvik (eds.), Kristiansand: Springer, pp. 188–194.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. 2011. “Action Design Research,” MIS Quarterly (35:1), pp. 37–56.

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., and Wang, L. 2012. “Modeling, Simulation, Information Technology & Processing Roadmap,” Washington.

Tao, F., Zhang, H., Liu, A., and Nee, A. Y. C. 2019. “Digital Twin in Industry: State-of-the-Art,” IEEE Transactiions on Industrial Informatics (15:4), pp. 2405–2415.

van der Valk, H., Haße, H., Möller, F., and Otto, B. 2021. “Archetypes of Digital Twins,” Business and Information Systems Engineering. (https://doi.org/10.1007/s12599-021-00727-7).

Vial, G. 2019. “Understanding Digital Transformation: A Review and a Research Agenda,” Journal of Strategic Information Systems (28:2), pp. 118–144.

Wache, H., and Dinter, B. 2020. “The Digital Twin – Birth of an Integrated System in the Digital Age,” in 53rd Hawaii International Conference on System Sciences, Maui, pp. 5452–5461.

Wache, H., and Dinter, B. 2021. “Digital Twins at the Heart of Smart Service Systems - An Action Design Research Study,” in 29th European Conference on Information Systems, Marrakech, pp. 1–17.

Wache, H., Möller, F., Schoormann, T., Strobel, G., and Petrik, D. 2022. “Exploring the Abstraction Levels of Design Principles: The Case of Chatbots,” in 17th International Conference on Wirtschaftsinformatik, Nürnberg, pp. 1–15.

Zheng, P., and Sivabalan, A. S. 2020. “A Generic Tri-Model-Based Approach for Product-Level Digital Twin Development in a Smart Manufacturing Environment,” Robotics and Computer-Integrated Manufacturing (64:101958), pp. 1–18.