PRODUCT LIFE CYCLE MANAGEMENT WITH DIGITAL TWINS FOR PRODUCT GENERATION DEVELOPMENT

Arnemann, Lars;
Winter, Sven;
Quernheim, Niklas;
Schleich, Benjamin

Product Life Cycle Management (PLCM), TU Darmstadt

ABSTRACT

Digital Twins are virtual representations of a product-service-instance and, as a technology, represent an important part of the realization of Industry 4.0. They manage data of the associat-ed product instance and can also have functions for simulation to achieve cost and resource savings while simultaneously increasing product quality. In this paper, a need for action for the implementation of a systematic approach for the returning of data of Digital Twins into the product design is identified and a methodology is developed as an answer. This methodology realizes an information management, which supports holistic data and information flows. It de-fines necessary steps for the implementation of data and information transport, starting from a data management up to information provision in product design. Based on a performed potential analysis for the identification of intended uses in the context of product design, the overall ap-plication focus is narrowed down to the development of new product generations to support the requirements development. The concept structure consists of Digital Twins, a data mining sys-tem for the transformation of data into information and a presentation system for managing the information provided.

Keywords: Product Lifecycle Management (PLM), Design engineering, Design methods

Contact:
Arnemann, Lars
TU Darmstadt
Germany
arnemann@plcm.tu-darmstadt.de

Cite this article: Arnemann, L., Winter, S., Quernheim, N., Schleich, B. (2023) ‘Product Life Cycle Management with Digital Twins for Product Generation Development’, in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.296

1 INTRODUCTION

For the first time, the Digital Twin is defined as a virtual representation of a physically existing product (Grieves, 2014). Since then, many academic publications with accompanying formulations of further definitions and use cases have been added (Lim *et al.*, 2020). Nowadays, it is an integral component in the list of relevant key technologies for Industry 4.0 (Sebastian Haag *et al.*, 2017). Furthermore, the combination of various existing technologies towards a new concept - the Digital Twin - enables efficiency gains on the one hand and new business models and, thus, new types of entrepreneurial value creation on the other (Samarjiwa *et al.*, 2020).

This paper defines Digital Twins as the virtual representation of a product (service) instance by the Scientific Society for Product Development (WiGeP, 2020). It also has an automatic bidirectional data exchange (Czwick *et al.*, 2020). In the WiGeP reference architecture, the Digital Twin has a digital master and a digital shadow (WiGeP, 2020). Based on the reference architecture, the technical design of the Digital Twin always depends on the defined application or use (WiGeP, 2020). Many different data types are managed by Digital Twins and map potential content on user and product behavior. From this, information can be derived, which leads to cost and resource savings and increased product quality when returned to product design. However, a systematic approach to returning data into product design is missing in the context of Digital Twins.

To realize this, a potential analysis of Digital Twins must first be used to identify a purpose for the concrete application of Digital Twin data in product design. In this context, appropriate system boundaries are defined. In this paper, the products under consideration are restricted to technical products produced in large numbers, regularly developed further (development of new product generations), and have a limited complexity. In particular, boundary conditions of an information management are to be included here so that a holistic data and information flow is realized. The concept will be elaborated to implement the systematic return of data from Digital Twins into the product design to ensure scalability, interoperability, expansibility, and fidelity (Schleich *et al.*, 2017).

The paper is structured as follows. The next section briefly overviews the state of the art in product data integration in product design. After that, the method to return data of Digital Twins is highlighted. In section four, the validation of this method is given, while section five provides a conclusion, followed by the acknowledgment and the references.

2 STATE OF THE ART

Riedelsheimer *et al.* (2018) developed a framework for the traceability of data managed by the Digital Twin. First, the Digital Lifecycle Twin is defined, which maps data from the product lifecycle over the entire product lifecycle (Riedelsheimer *et al.*, 2018). Based on this, the process steps of data acquisition, data processing, data transfer to the Digital Twin, and storage and management must be performed to transfer the data to the Digital Lifecycle Twin. This is followed by automated data analysis, which is envisioned as knowledge generation (Riedelsheimer *et al.*, 2018). In the next step, the knowledge will be returned as decision support and transferred into product improvements in the final step (Riedelsheimer *et al.*, 2018).

Continuing, Tao et al. (2019) developed an extended framework for product design driven by the Digital Twin. The goal is to support the development of new product generations and to structurally integrate Digital Twins into the product design process (Tao *et al.*, 2019). The concept is based on the key technologies Internet of Things (IoT), Cyber-Physical Systems (CPS), and Big Data Analytics (Tao *et al.*, 2019). IoT networks enable data generated by CPS to be transmitted and thus managed by Digital Twins. Big Data formulates the technology needed to analyze the large volumes of data generated by a fleet of CPSs over their lifecycle to generate value (Tao *et al.*, 2019). The Digital Twin manages various data types (Tao *et al.*, 2019). To apply Digital Twin data in product design, the development process must be divided into the phases of clarifying the task, conception, design, and elaboration, according to Pahl *et al.* (2007), and the V-model must be integrated into these phases (Tao *et al.*, 2019). Depending on the development activity, different data types are relevant. Figure 1 visualizes the approach of Tao *et al.* (2019) and includes exemplary data sources depending on the development phase (Tao *et al.*, 2019).

For the use of the returned data in product design, the authors formulate four steps from the data science perspective (Tao *et al.*, 2019). First, data collection is defined as generating and recording the raw data (Tao *et al.*, 2019). Then, data integration is needed to merge the different data formats (Tao *et al.*, 2019). The next step is data cleansing, which cleans data from sources of error (Tao *et al.*, 2019).

Finally, the data mining process takes place, extracting information from the raw data via further processing and analysis that is relevant to product design (Tao *et al.*, 2019).

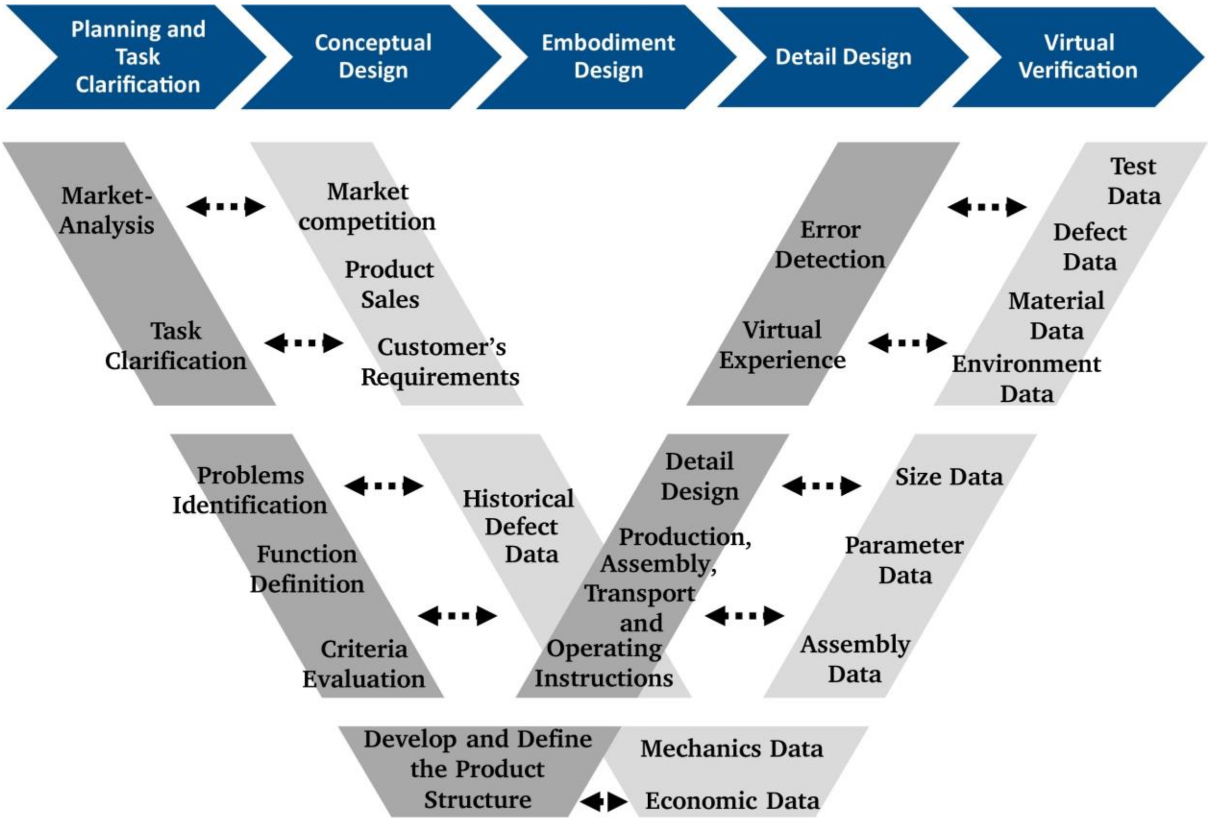

Figure 1 - Framework for inclusion of Digital Twins into product design (adapted from Tao et al 2019)

The information logistics aspect in the sense of information management remains insufficiently described. Information logistics describes the transmission of the right information content, to the right extent, at the right recipient location, at the right time, and in the right quality. There is a vague answer to how efficient information logistics can be realized in the context of the numerous forms of Digital Twins and the data they manage. A central question is how data from Digital Twins reach the product developer through a transport process and how the developer accesses it. The starting point for this is the question of the type and scope of concretely existing information requirements (correct information content, correct time of provision) on the part of the product developers for which Digital Twin data offer a relevant source of information. From this, the support of the product developer in coping with the task or problem is realized. In addition, the question must be answered about how the information packages to be provided (quality, location, and scope) must be designed to realize holistic logistics.

3 METHOD TO RETURN DATA OF DIGITAL TWINS INTO PRODUCT DESIGN

The Digital Twin manages data directly related to the associated product instance. To return data from Digital Twins into product design for product generation development n+1, shifting the observation level from the instance to the product generation level is advised. Furthermore, automated analysis of the aggregated data is necessary to regularly update the results because only by shifting to the product generation level instance-independent product characteristics can be identified. In addition, the state of the art shows many possible applications of data of Digital Twins for traceability in the context of requirements development. It must also be taken into account that organizations develop many different products. As a result, the methodology must map different service cases in connection with a product.

https://doi.org/10.1017/pds.2023.296 Published online by Cambridge University Press

On the one hand, activities that specify the transport of data and information in dependencies of the selected service case and the product must be defined. On the other hand, resources or tools must be defined that provide the necessary information to implement the process in the form of a structure and thus realize the provision of information. A Structured Analysis and Design Technique (SADT) diagram conceptualizes the methodology (Ross, 1977). The diagram allows the representation of a sequence of necessary activities as a black box, which are necessary for the realization of a return, with a simultaneous representation of the data and information flow, as well as resources to be included.
Figure 2 depicts the methodology's traceability concept in a SADT diagram. The activities shown are subdivided into data identification, data retrieval, data and information processing, and information provision.

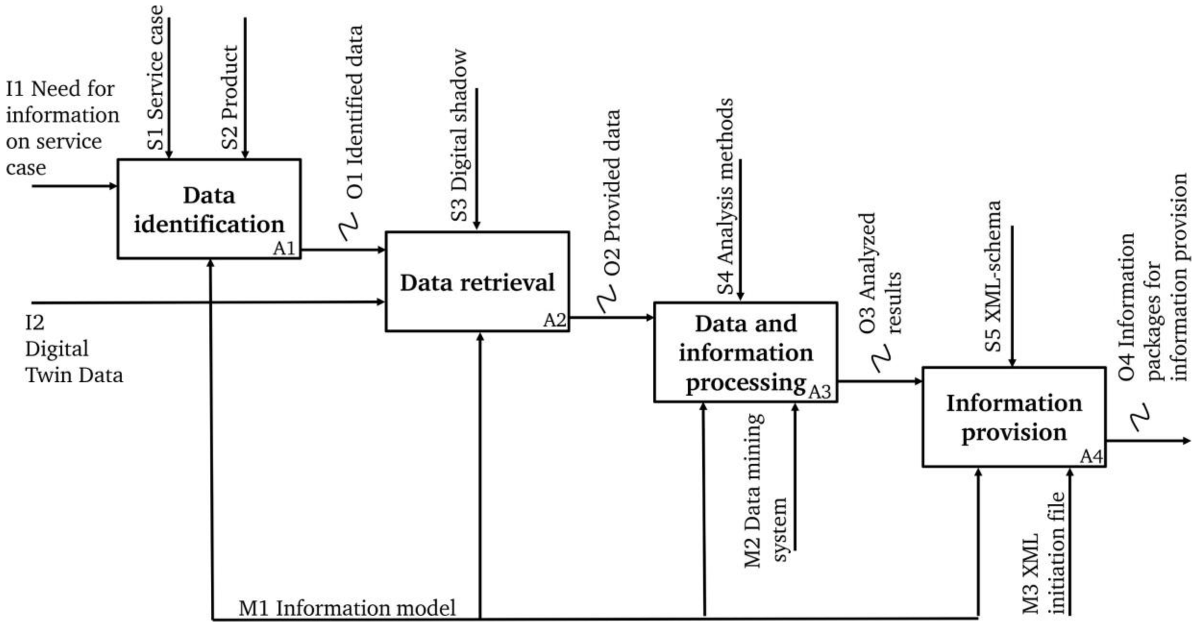

Figure 2 - SADT diagram for the conceptual design of the method (own illustration)

Integrating the methodology into the system landscape results in a holistic flow of data and information, from instance-based data generation to management and information presentation to the use of information by the product designer. This results in the realization of a complete data lifecycle and the implementation of an infrastructure for the creation of information logistics and context-based information provision. Figure 3 illustrates this classification. Starting with Digital Twins, the digital shadow manages all data of Digital Twins at the level of the product instance. Data based on the provision of information within the scope of defined service cases for a product generation n can be retrieved via data identification. These are transformed via the data mining system into information or knowledge relevant to the product design for the requirement development of the product generation n+1.
Finally, results are supplemented with additional information and documentation and transferred to the presentation system as an Extensible Markup Language (XML) file. XML is a markup language that can be interpreted, read, and created by users and machines alike (Rothfuss and Ried, 2001). It enables a hierarchically structured representation of the information and data to be provided and also represents a standardized data format for the exchange and transmission of data (Rothfuss and Ried, 2001). The necessary data and information are structured by an information model managed in a separate storage system. The management of the provided information is realized in the presentation system and made accessible to the product developer via a visualization. On a higher level, an assistance system is realized through integrating and implementing the methodology in software, which supports the product designer in the product generation development n+1.
The implementation of the data and information transport requires the modeling of a process that defines the steps of a return. A Unified Modeling Language (UML) activity diagram is used to develop and represent this process model, which allows the representation of a process or the modeling of the dynamic behavior of a (software) system by structuring the flow of data and information using formalized elements.

https://doi.org/10.1017/pds.2023.296 Published online by Cambridge University Press

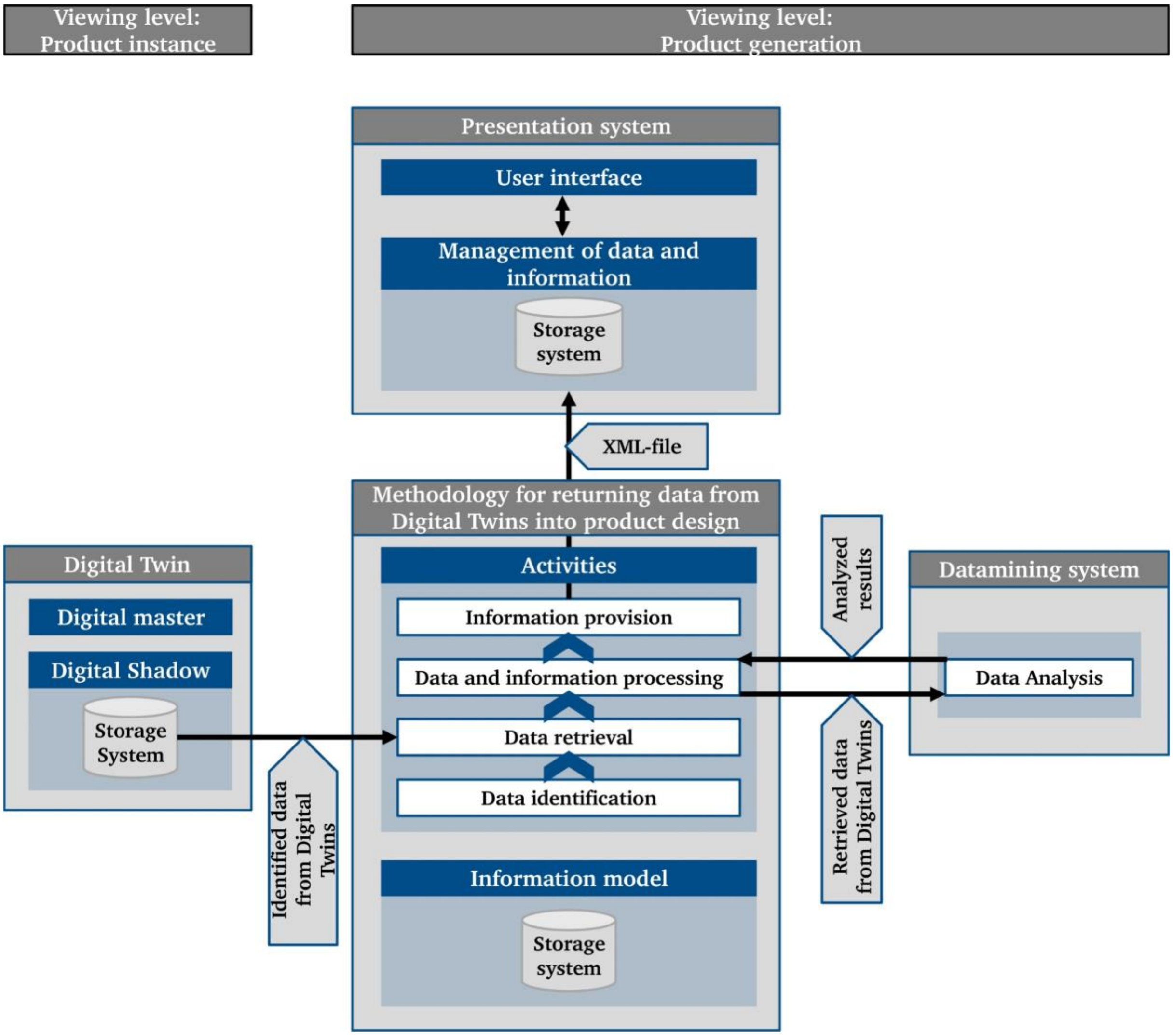

Figure 3 - Concept representation in the system landscape (own illustration)

Figure 4 depicts the process model in its entirety. Here, the process model is based on the activities of the conceptual representation of the method, which are elaborated in the activity diagram. As information provision and design of efficient information logistics, a push principle is provided to the method. This means that information is always provided to the product designer in an automated and contextualized way while there is a regular check on the timeliness of the results (Schuh *et al.*, 2020). This means that the product designer is not responsible for generating results and, thus, providing information. This is done in the sense of the design of assistance systems because an integration of the assistance into the course of action is aimed (Schuh *et al.*, 2020). A requirement for the traceability of data of Digital Twins and, thus, for implementing the process model is that the necessary data and information for a product and service case are already structured and mapped via an information model. First, the actions and activities are divided according to system affiliation. Therefore, the subdivision is done analogically to the conceptual representation.

The *product designer*, the *presentation system*, the *developed method to return data of Digital Twins into the product design*, the *datamining system*, and *Digital Twins* are differentiated. The starting point of the process model is the initial generation or the check of already provided results for updating. The starting point for realizing the push principle is the method itself. Related to software realization, an implementation can be realized by setting up an execution frequency. The execution frequency specifies the (temporal) interval in which the program is executed.

As an illustration, an implementation is done in an execution frequency of number X per interval. An organization can carry out the development of a large number of different products. In order to return data of Digital Twins, the product must first be selected for which a new product generation n+1 is to be developed, and information must be made available.

https://doi.org/10.1017/pds.2023.296 Published online by Cambridge University Press

This selection is based on the product information objects the information model represents. The product information objects represent all products for which service cases are identified. Once the product has been selected, the selected service case action must be performed. From the derivation and description of the service cases, it is clear that each product can have many concrete service cases for the return of Digital Twin data. The specific service case for which traceability is to be performed must be selected. The source of information is the service case objects of the selected product information object.

The update frequency attribute of the service case object is used to specify the frequency with which the results are checked to ensure that they are up to date. Based on the selection of a specific service case depending on a product, a target/actual comparison is performed, which results in the process's continuation (update required) or termination (update not required). The check of the update frequency is necessary to prevent avoidable operational burdens on the data mining and presentation system.

In this step, all higher-level products from whose instances data are provided for information provision must first be identified. Likewise, various data are required depending on the selected service case. In addition, this differentiated data is managed by one to n different Digital Twins or their digital shadows. For further processing, these have to be retrieved. In this action, the required parent products, its data, and the concrete administration addresses must first be identified. This information is retrieved from the data origin objects of the service case object. The result after this step is an identification of all data sources needed to return to product design in the context of a specific service case and product.

Based on the identification, data from Digital Twins will be retrieved via the required interfaces and suitable communication technologies and provided or transferred by the relevant Digital Twins. As a result, the required data is available for further processing.

The first step in further processing is to check whether there is a need for data analysis, i.e., updating the results provided based on changes in the data basis. The starting point is, first of all, the basic check of whether results have already been generated and thus made available for the product concerned. Then, the results object manages the necessary information. If no result is available, more processing takes place.

If a result has already been provided, its actuality must be checked in the context of this action. The result object represents a quantitative key figure for this purpose. The key figure represents a comparative value against which the properties of the provided data are to be checked. This varies from a time interval or duration, which must be reached since the last update, up to concrete exceeding threshold values (e.g., the temperature is 20% above the last average). If the check reveals that the results already returned are up-to-date, no further processing takes place, and the process is terminated at this point. The product designer continues to use the results already provided.

The subsequent further processing takes place in a parallelization. First, an XML initiation file must be created for information provision and result management. Then, starting from the XML initiation file, additional information will be added that enables the product designer to contextualize and thus facilitates or assists in interpreting the results provided. The goal is to minimize the probability of erroneous conclusions from the provided information by the product designer by adding additional information. The addition of requirements for the product generation n is to be particularly emphasized at this point. This results in a direct contextualization of the specified and real effective variables. Here, the adding is to be made by the method itself in order to minimize additional activities for the information procurement by the product designer and thus sources of error. The information is represented in the requirement object. Further additional information for contextualization is also to be derived from the information model. Here, the product meta information object, the service case object, and the data objects are to be used as direct sources of information.

Moreover, documentation must be added to the instantiated XML file for additional information. The documentation creates a basis for a transparent and traceable flow of information and is ultimately the basis for data-based decisions. Here, documentation includes meta information about the data and associated analysis methods. This information is presented in the analysis and data objects of the service case. The structure of the XML file represents a defined XML schema and thus allows the standardized provision and management of information packages. The structure of the XML schema is directly oriented to that of the information model. An additional information block is added in the XML file, which maps the analysis results.

https://doi.org/10.1017/pds.2023.296 Published online by Cambridge University Press

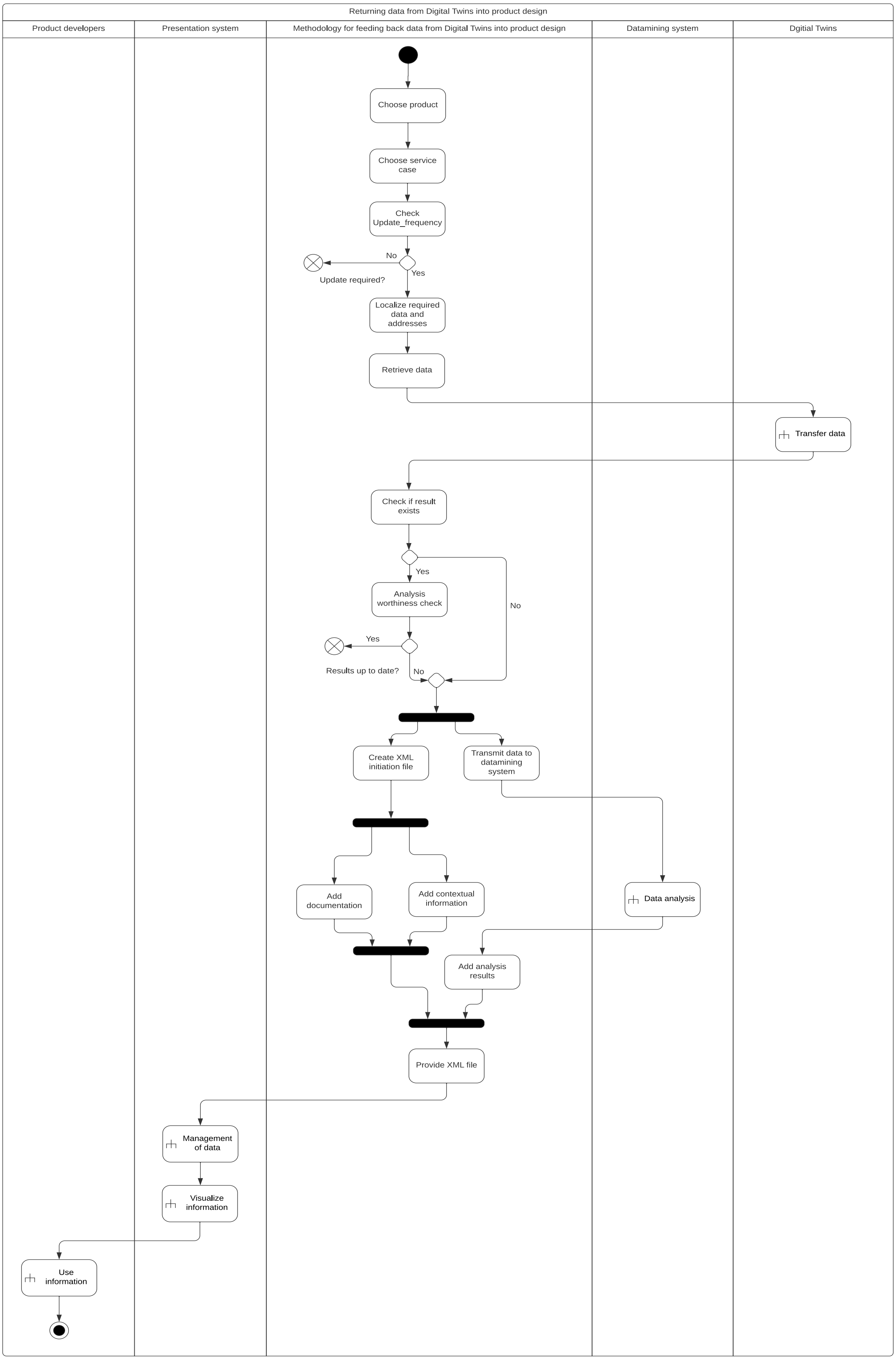

Figure 4 - Process model for returning data from Digital Twins into product design (own illustration)

https://doi.org/10.1017/pds.2023.296 Published online by Cambridge University Press

In parallel, the data retrieved from the relevant Digital Twins are forwarded to the data mining system. For this purpose, the data analysis method always depends on a combination of product, service case, and existing data.
As a result, the data mining system performs the data analysis. The analysis results of the data mining system are then transferred to the methodology via suitable interfaces. The functionality of the data mining system is the same as described. According to the XML schema, the results will be attached to the initialized XML file. In the following step, a result object shall be created that includes documentation of the calculation time, the key figure for checking the update, and the number of different result versions (number of updates of the result). In addition, this information shall be attached to the XML file. This results in an information package forwarded to the presentation system for management via the required interface.
The presentation system manages the transmitted information packages and presents them visually. Finally, the product designer accesses the information provided via the user interface of the presentation system and uses it for the application.
By repeating the process, all managed service cases for the superordinate products have to be run through one after the other, and thus a return of Digital Twin data takes place for all service cases and products structured via the information model.

4 VALIDATION AND VERIFICATION

The validation is based on a simulative approach and does not aim to complete the development process for the product generation n+1 of the original Prusa i3 MK3S+ printer. Only a Digital Twin of a single product instance exists, so it is impossible to obtain sufficient information from a performed data analysis to realize support for a requirements development for product generation n+1. Thus, the focus of the validation remains on representing an operational data and information transport process.

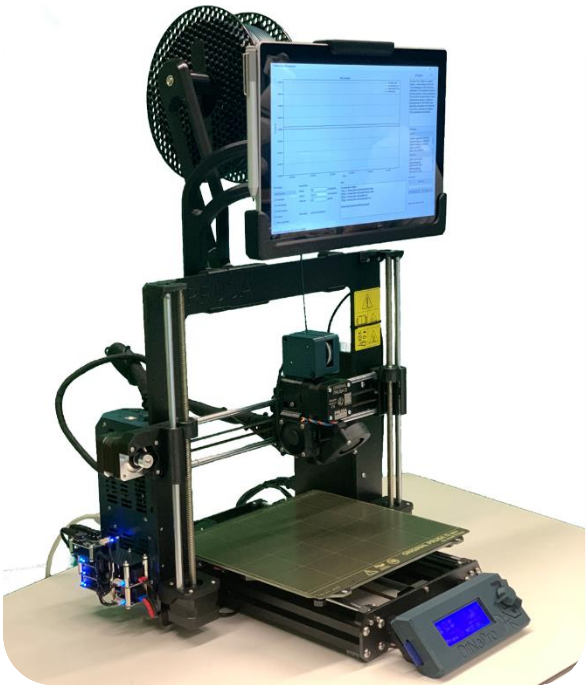

Figure 5 - Modified Prusa i3 MK3S+ (own photo)

The Original Prusa i3 MK3S+ 3D printer is identified as a suitable product for validation. This is a 3D printer that uses the Fused Deposition Modeling (FDM) process, which is part of additive manufacturing. As part of a project, sensor technology is being added to the product instance (Figure 5). The extension allows the measurement, transmission, and storage of product instance-specific data. In addition, electrical power measurement of the components heating bed and nozzle will be added and at the power supply unit to record the required total electrical power (input side). An encoder is also retrofitted to measure the material consumption between the filament roll and the nozzle.

https://doi.org/10.1017/pds.2023.296 Published online by Cambridge University Press

By implementing the developed methodology, automated data transport is carried out, from the Digital Twin, via the data mining system to the presentation system. Here, the process steps of the methodology are carried out one after the other. First, data transferred in dependence of the product Original_Prusa_i3_MK3Splus_printer are identified and retrieved from the Digital Twin. The retrieved data is then forwarded to the data mining system. Next, an implemented Python program is used to perform the evaluation. Finally, an XML file is initiated, results are contextualized using additional information, and documentation is added. Figure 6 shows an extract from the instantiated XML file that provides the information.

```
<data_description>
  <identification_number>MK3S+</identification_number>
  <product_designation>Original_Prusa_i3_MK3Splus_Printer</product_designation>
  <data_designation>PowerHeatbed_calc</data_designation>
  <data_description>power of the heatbed</data_description>
  <unit>watt</unit>
  <data_type>double</data_type>
  <life_cycle_phase>use</life_cycle_phase>
</data_description>
```

Figure 6 - Extract from XML file for information provision (own representation)

Based on the validation, previously defined requirements for the method are checked through verification. The basic function is to be considered fully fulfilled. In the course of the work, service cases are first described. These represent a concrete information need of the product developer for the execution of the development activity or the specification of requirements for the product generation n+1. The service cases specify the intended use of the returned data. Based on the service cases, a method is developed as a process model. Likewise, the data timeliness is to be considered as full compliance.

The method enables an automated return of data of Digital Twins and integrates several check actions for the actuality assessment. Likewise, additional information is transmitted to increase the information quality for contextualization.

Furthermore, documentation is created. From this follows the realization of transparent, data-based decisions and thus ensures the traceability of the data origin. The process model is developed with extensibility in mind. The automated transmission of the required information takes place by placing the method in the context of the data mining system, the presentation system, and the selected push function. Via the presentation system, the information is visualized so that accessibility is guaranteed for the product developer, and the set requirement is considered fully fulfilled. The requirement for use is thus also considered to be fully met.

Due to the choice of a push function, the method cannot fulfill the security requirements. Nevertheless, the concept integrates the approach via the design of the presentation system.

The core of the developed methodology is presented via an activity diagram. Diagrams of the UML support an object orientation and thus a convertibility as software. Consequently, the set requirement for the representation of the methodology is to be regarded as fully met.

In summary, the requirements for function, security, and representation can be regarded as partially fulfilled so that, in conclusion, development success follows from the evaluation.

5 CONCLUSION

In the course of the work, a method to return data of Digital Twins into product design is developed with the goal of developing a requirements basis for the product generation n+1. The method realizes a service case-based return of Digital Twin data into the product design. Service cases specify a concrete use case or purpose of data of Digital Twins for application in product design. As a realization, an UML activity diagram is developed to model the necessary processes for a return.

Product design is part of product creation. This additionally consists of product planning and production. The product planning phase includes the development of business models. The product lifecycle management with Digital Twins for product generation development can be applied to identify new business areas. As an illustration, via an analysis of Maintenance, Repair and Operations (MRO) reports and the identification of frequent machine failures, a customer-specific, additional service for condition monitoring or predictive maintenance can be offered. This creates a further potential for value creation during the utilization phase.

The method fits into the concept structure consisting of Digital Twins, a data mining, and a presentation system. In the course of the work, different concrete use cases or service cases are identified based on the heterogeneous forms of Digital Twins. The design of the data evaluation method depends on the combination of existing data, product, and service case. As a result, the increasing integration of various service cases gives rise to many developed methods for processing Digital Twin data for returning into product design. This requires efficient knowledge management, which first manages the aggregated knowledge from methods, data, products, and service cases. Based on this, the knowledge from implemented analysis methods is to be reused, and thus the reduction of development times for new analysis methods is to be realized.

Another outlook is the presentation system, which acts as an interface to the product developer. Therefore, the design of the user interface has to be implemented and adapted to the context of use to achieve an optimal provision of information. In this context, the activities, tasks, and problems of product development differ greatly from those of production (e.g., manufacturing) and use. This results in the need for elaborating concrete and reusable design parameters of a graphical user interface to present information and the associated realization of decision support efficiently.

ACKNOWLEDGMENTS

This research is funded in the DiNaPro-project by the German Federal Ministry of Education and Research (BMBF) and implemented by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the content of this publication.

REFERENCES

Czwick, C., Martin, G., Anderl, R. and Kirchner, E. (2020), “Cyber-Physische Zwillinge”, *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, No. s1, pp. 90–93.

Grieves, M. (2014), *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*.

Lim, K.Y.H., Zheng, P. and Chen, C.-H. (2020), “A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives”, *Journal of Intelligent Manufacturing*, Vol. 31 No. 6, pp. 1313–1337.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung; Methoden und Anwendung*, 7. Aufl., Springer, Berlin, Heidelberg.

Riedelsheimer, T., Lindow, K. and Stark, R. (2018), *Feedback to design with digital lifecycle-twins: literature review and concept presentation*.

Ross, D.T. (1977), “Structured Analysis (SA): A Language for Communicating Ideas”, *IEEE Transactions on Software Engineering*, SE-3 No. 1, pp. 16–34.

Rothfuss, G. and Ried, C. (Eds.) (2001), *Content Management mit XML: Grundlagen und Anwendungen*, *Xpert.press*, Springer Berlin Heidelberg, Berlin, Heidelberg.

Samarjiwa, M., Lindow, K., Salomon, D. and Stark, R. (2020), *Digital Twin Readiness Assessment: Eine Studie zum Digitalen Zwilling in der fertigenden Industrie*.

Schleich, B., Anwer, N., Mathieu, L. and Wartzack, S. (2017), “Shaping the digital twin for design and production engineering”, *CIRP Annals*, Vol. 66 No. 1, pp. 141–144.

Schuh, G., Anderl, R., Dumitrescu, R., Krüger, A. and Hompel, M. ten (Eds.) (2020), *Industrie 4.0 Maturity Index: Die digitale Transformation von Unternehmen gestalten*, Update 2020, acatech - Deutsche Akademie der Technikwissenschaften, München and Berlin and Brüssel.

Sebastian Haag, Klaus Schützer and Eduardo Zancul (2017), “Digital twin technology – An approach for Industrie 4.0 vertical and horizontal lifecycle integration”, *it - Information Technology*, No. 3, pp. 125–132.

Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Guo, Z., Lu, S.C.-Y. and Nee, A.Y.C. (2019), “Digital twin-driven product design framework”, *International Journal of Production Research*, Vol. 57 No. 12, pp. 3935–3953.

WiGeP (2020), “WiGeP-Position Paper - Digital Twin”, available at: https://b7s1f6.n3cdn1.secureserver.net/wp-content/uploads/2022/05/Final_WiGeP_Positionspapier_Digital_Twin_englisch.pdf (accessed 23 February 2023).