INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.158

Approach for Developing Digital Twins of Smart Products Based on Linked Lifecycle Information

T. Eickhoff ✉, S. Forte and J. C. Göbel

Technische Universität Kaiserslautern, Germany

✉ eickhoff@mv.uni-kl.de

Abstract

The ongoing digitization of engineering processes and the increasing prevalence of smart products create possibilities for new business models and services. Digital twins enable the collection of all required data about a smart product in order to make these possibilities a reality. This paper describes a flexible approach towards a digital product twin that is tightly integrated with existing product models while being lightweight and easy to integrate with existing IT solutions.

Keywords: digital twin, product lifecycle management (PLM), digital thread, process improvement, data-driven design

1. Introduction

The ongoing digitization of engineering processes and the increasing prevalence of smart products create possibilities for new business models and services (Porter and Heppelmann, 2014). However, implementing solutions targeting these possibilities offers new challenges. The data generated from each individual instance of a smart product have to be managed and relevant information for new emerging use cases has to be available at the right place and time in sense of its lifecycle. Combining this so-called "digital twin" of the smart product instance with the required context information becomes increasingly difficult when a holistic view across the product's lifecycle is required. In a realistic scenario, traditional product data, as well as new data from smart products (e.g. usage-data), are each typically stored across several, not necessarily compatible and open-standard-based information technology (IT) solutions. This paper describes a lightweight and open (in terms of standards) approach towards the introduction of digital twins in the context of existing IT infrastructure and systems. The focus of our approach is the ease of introduction in an existing IT context.

2. Lifecycle Information Management in the Context of Digital Twins

One of the first uses of the term digital twin was by NASA, which introduced digital twins for integrated, multi-physical and multi-scalable simulation tasks in 2010 as part of a technology roadmap (Shafto *et al.*, 2012; Savarino, 2019). From today's perspective, a digital twin is a virtual representation of an existing or future real object, including its structure, properties and behavior (Schroeder *et al.*, 2016; Kuhn, 2017). In principle, these can be several kinds of real objects. In the engineering context, relevant objects are (smart) products and services including the surrounding context they are embedded in (e.g. area of application, framework and stakeholders) (Forte *et al.*, 2021), prototypes or simulation models as well as production machines and plants (Stark *et al.*, 2017). Digital product twins in a narrower sense are based on smart products with a physical component. The following text will assume this narrower context of the term. Digital product twins contain a consistent set of functionally integrated virtual sub-

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

models and product lifecycle data of the corresponding smart product instance concerning the specific application purpose.

In the area of development of smart, interdisciplinary products, the term "digital model" (sometimes also called "digital master") describes the generated data by various disciplines during all phases of the development process. The digital model is typically the sum of several sub-models from different domains, not necessarily stored in a single place. Managing an interdisciplinary digital model requires an appropriate methodology. One approach to this problem comes from the field of Model-based Systems Engineering (MBSE).

In comparison to traditional systems engineering, MBSE uses a coherent model of the system in development and therefore offers a consisting set of methods and processes supporting holistic systems design and analysis (ISO 15288, 2015). Often starting in the conceptual design phase (e.g. early roadmapping phases), MBSE supports also the early system verification and validation activities especially in the requirements engineering and system design lifecycle phases and is furthermore a great communication vehicle among all engineering-involved disciplines and stakeholders to create a shared understanding of the system of interest (SoI). This also reduces the perceptions of system complexity (Friedenthal et al., 2014). The system model also serves as a basis for further verification and validation processes and is then being considered as a jumping-off point, for downstream, discipline-specific development phases (Friedenthal et al., 2014) described the following three elements furthermore as necessary for describing a system within a model. A design method for system specification, a modelling tool and a modelling language. The language in this context is used to describe the systems architectures by integrating views on the systems requirements, functions, behavior, structure and also to support further decisions in the engineering process based on these architectures (Alt, 2012). A great benefit of such comprehensive system models is also the possibility to explicitly define links for traceability purposes between different modelling elements. For example, to track the satisfaction of requirements in the solution space of the modelled system (Bernard, 2012). These links form the so-called “digital thread”, which ensures traceability in the digital model and digital twin throughout the complete lifecycle. From an MBSE perspective, all these links exist in a coherent system model. In a typical engineering IT context, however, these relationships typically exist as explicit or implicit connections between data objects stored in several different IT systems. Making the required data available at the right time and place is often a non-trivial problem especially from an engineering IT perspective. Several approaches for managing this complexity are presented in the following section.

Data forming the digital model and/or digital twins is typically stored in several IT systems across different departments within a company or even different companies (e.g. in extended, cross-company value creation networks). Bringing the data together for various engineering processes requires data exchange, synchronization or a decentralized approach.

Currently, a variety of solution approaches for cross-system data management exist, such as (Lentes et al., 2012; Abramovici et al., 2016; Abramovici and Herzog, 2016). Thus, data stored in different systems is partially synchronized with other systems, which allows straightforward access for users from the affected systems, but by definition introduces some redundancy. This violates the "single source of truth" approach (Bergsjö et al., 2006) and carries the risk of inconsistent states, i.e. contradicting versions of the same data stored in different systems.

This contrasts with a decentralized approach, where data remains in the respective source systems and is retrieved as needed. This results in longer access times (because data has to be fetched from its source) but eliminates redundancies, which in turn reduces overall storage consumption and the potential for inconsistencies (Bergsjö et al., 2006). One example of a decentralized approach is the Open Services for Lifecycle Collaboration (OSLC) (Amsden and Speicher, 2021). OSLC defines a minimalist core vocabulary and extensions for several application domains. Systems communicate using this unified vocabulary over web interfaces, following best practice approaches like Linked Data and the Resource Description Framework (RDF).

Mapping - the linking of data objects - can be done either manually or (semi-)automatically. An example of an automatic approach is described in (Zeimetz and Schenkel, 2020). Apart from mapping data by creating links, ontologies enable semantic interoperability by providing common information

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

models. The creation of ontologies is a highly specialized field with different methods, but they require domain knowledge as well as expert knowledge (Hildebrandt et al., 2018). The need for specialized knowledge increases with the complexity of the ontology. One can distinguish between lightweight and heavyweight ontologies. The former are informal or semi-formal descriptions. Software engineering approaches such as the Unified Modeling Language (UML) or entity-relationship diagrams are suitable for creating lightweight ontologies. Heavyweight ontologies also contain these taxonomies but add axioms and constraints to provide deeper meaning (Corcho et al., 2006).

3. Concept

Managing smart product-based digital twins requires an understanding of the specific context where the digital twin will be embedded in and furthermore used. Based on this understanding, an adaptive IT technical solution has to be developed and implemented in the existing IT infrastructure. For this purpose, a modelling framework for digital twins of smart products and a series of software demonstrators that show the potential of the envisioned solutions has been developed. The high-level idea is as follows: First, develop an understanding of the desired application scenario. This can be an informal description, or follow a more formalized methodology, e.g. a Business Model Canvas. Then, define several design components of a conceptual solution for this scenario in terms of a digital twin and the underlying platform with its interfaces to external systems (our framework for this step is described in section 3.1). Finally, implement a (prototype of) the technical solution. At this stage, the broad notion of a digital twin has already been detailed according to the developed concept. We provide an example process for the implementation of Product Lifecycle Management (PLM) centric derivation of a digital product twin. Finally, we will present several prototypes that illustrate specific aspects of the design process.

3.1. Digital Twin Modelling Framework

The design of a digital product twin cannot be seen as an end in itself. It needs to address the requirements put forth by a specific application scenario (the scenario can take the form of clearly defined business use cases, but can also contribute to broader defined future application areas). This application-focussed view lead to the development of the Kaiserslautern Digital Twin Modelling Framework (KDTM) (Göbel and Eickhoff, 2020). KDTM assists the design of specific digital product twins by collecting answers to questions about different components of the desired field of application, leading to a clear concept of the specific digital product twin before the development of an actual technical solution.

The scope of this design framework is to methodically clarify basic decisions about the planned conditions of use and the design characteristics of company-specific, digital product twins in a conceptual planning phase before developing technical solutions. The methodology and core areas of the framework are depicted in the left half of Figure 1. The design framework is structured in the following three component areas:

Before detailed planning can begin, the "overarching conceptual design components" must be considered. These include the intended field of application, with immediate or potential application for new business models (e.g. reconfiguration services, assistance functionalities for product use, maintenance etc.) or for optimizing processes during the product's development (e.g. data analyses for requirements development, simulations and product validations). Another factor are the referenced processes, i.e. product lifecycle processes on which the application scenario of the product twins is focused and interfaces to adjacent product lifecycle processes (e.g. sensor data integration into lifecycle models for predictive maintenance processes. Similarly, the twin's physical counterpart needs to be defined. This could be entire product ecosystems (e.g. system of systems), products, specific product components, assemblies, or product-integrated smart services. Finally, the twin's envisioned lifecycle phase integration should be described in terms of relevant product lifecycle phases (product development, production, use, reconfiguration, end of life) in which product twins are created and used in terms of information technology, as well as their information technology continuity (digital thread).

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

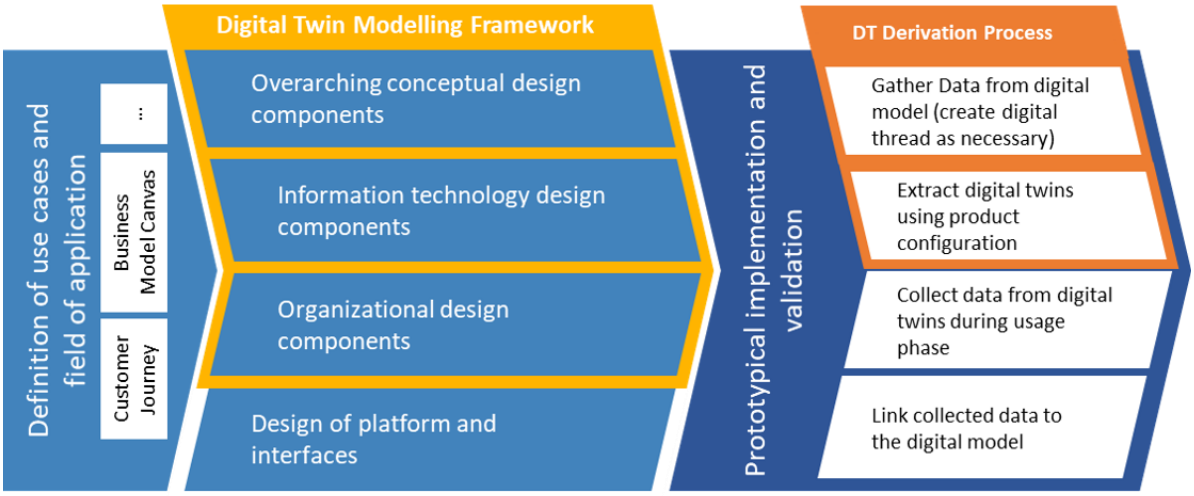


Figure 1. Process for the design and implementation of digital twins

After the general components are defined, the framework focusses on the "information technology design components". The digital twin's interoperation capabilities describe the connection of the digital product twin to cross-vendor platforms, to vendor-specific platforms, connection with other digital twins of similar or different products or with non-digital twins without a product focus (also cross-company and cross-industry sectoral). The communication between digital and real twin has to be defined in terms of the required communication infrastructure for synchronization of the twins (Internet of Things (IoT) driven platforms, hardware and software-based connectivity of real product twins, communication infrastructure (real-time capability, 5G, ...). The digital twin's envisioned model and data scope includes relevant model formats, data formats, engineering data (results of modelling-simulation and testing processes as well as data on performed service work), sensor and environmental data of the real product twin, as well as relevant external smart services. The model and data management can take the form of a centralized or federated management of the components of digital product twins across company boundaries in the cloud, within the product provider company (across several IT systems, e.g. PLM systems, failure data bases and IoT platforms, …) or embedded in the product (e.g. digital service booklet for cars). Data collection can be a combination of automated collection via sensor technology inside the product (system of interest) and outside the product (more ecosystem related) in question, automated collection in the engineering process (e.g. maintenance documentation, automatically collected customer feedback etc.) or manual data entry with or without a defined process, driven by individual actors.

Finally, the "organizational design components" complete the description of a specific digital twin design. These start with the expected user groups, i.e. the cross-company responsibility for model and data generation, management and use by user groups in the relevant product and engineering lifecycle phases and disciplines, including the necessary organizational interfaces. Model and data ownership can be defined in terms of ownership rights for models and data of the digital product twin and their implementation in the cross-company use of digital twins, if applicable, as well as business offers for the full or partial use of digital product twins. Finally, the role and authorization management encompasses access rights, approvals and required views of data and models across company boundaries.

After considering these three broad areas (conceptual, IT and organizational design components), the output of the framework is a high-level view of the envisioned scenario and the corresponding digital twin. Based on this view, the next step is the design of a suitable platform and interfaces to all intended data sources. This includes a survey of existing data sources that should be connected, as well as a first draft of a complete high-level architecture that includes all software and hardware components that need to be developed. This architecture acts as a starting point for the first technical implementation. We provide a sample process for a specific subset of digital twins in the form of the digital twin derivation process described in the following section.

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

3.2. Derivation of Digital Product Twins

The huge solution space of the framework described in the previous section makes it impossible to describe a single approach towards the implementation of arbitrary digital twins. However, some general observations can be made and, with some additional assumptions, be extrapolated to an approach for a relatively tightly integrated digital twin for a smart product. This approach is depicted in the rightmost part of Figure 1. The digital twin generally depends on the digital model, which contains context information that is needed to make sense of the data a smart product generates. This can include information about possible product variants and information relevant to the specific configuration of the product instance in question. For a reasonably self-contained digital twin in the context of maintenance and repair processes, this information should be available to the service partner performing these processes. In some cases, the processes might even change the product's configuration, which in turn needs to be represented in the digital twin. Based on these ideas, we propose the following process:

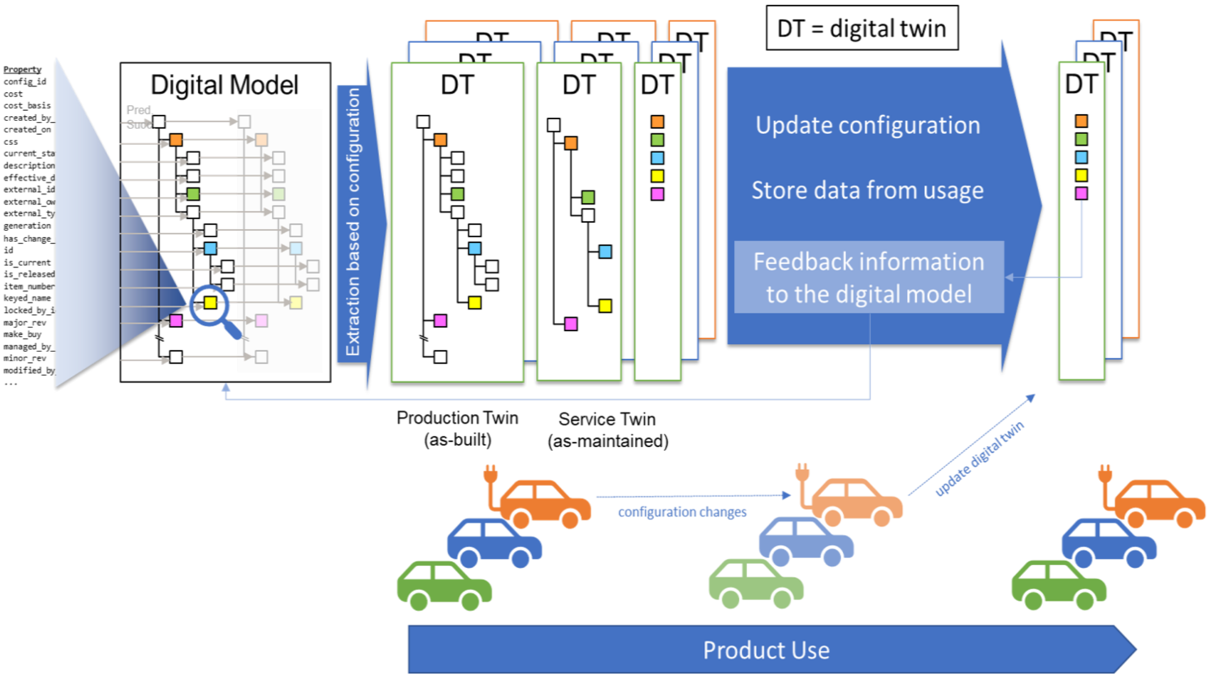


Figure 2. Data flows between digital model, product twin and product

After the initial design of the basic parameters of the envisioned digital twin, all necessary data from the digital model must be collected (this can be achieved through different processes described in the following subsections). The data is then filtered such that it only contains information relevant to the product instance's current configuration (as illustrated in Figure 2). Based on the parameters chosen during the twin's design, the twin can then be stored in an appropriate location. Different subsets of the data can be used for use-case-specific digital twins. If this location is separate from the digital model, an extract of the model's relevant information has to be stored with the twin. If the twin is tightly coupled to the digital model, it is also possible to store links to the original data.

The twin collects data from the product use phase. If the product's configuration changes, these changes need to be mirrored in the digital twin. Changes in the twin can reference back to objects in the digital model. This information can be used in the design of future product generations. An example of the interplay between digital twin and digital model will be described in section 4.

From a theoretical standpoint, a single unified digital twin is desirable. However, different use cases with different needs and constraints such as data ownership may require several partial digital twins for different purposes. In particular, different levels of hierarchy (tree-structure vs. flat list) or levels of detail and amounts of integrated data can be achieved for different use-case-specific twins.

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

3.3. Metadata-based Management of Lifecycle Information

As mentioned in the previous section, the derivation of the digital twin assumes that all necessary data from the digital can be collected and, depending on the twin's parameters, copied to or linked from the twin. This is easy to say, but difficult to realize in a realistic scenario, where the digital model is distributed over several incompatible IT systems. As mentioned in section 2, several approaches exist to realize the digital thread, which connects all these sources.

Our approach is a metadata-based integration, i.e. the actual data resides in the source systems, while the connections are based in a separate system that contains a graph-based overview of all data objects and their connections (including the digital thread). The objects in this graph contain no copied data, but rather meta-information on what type of object it is and where it can be found. Information is then retrieved on demand over web interfaces. The metadata graph works similar to existing ontology-based approaches or knowledge-graph-based integration approaches but avoids the complexity of more detailed ontologies. Additionally, the approach minimized duplication of data, such as introduced by a full-fledged integration solution. The envisioned process for this approach is depicted in Figure 3.

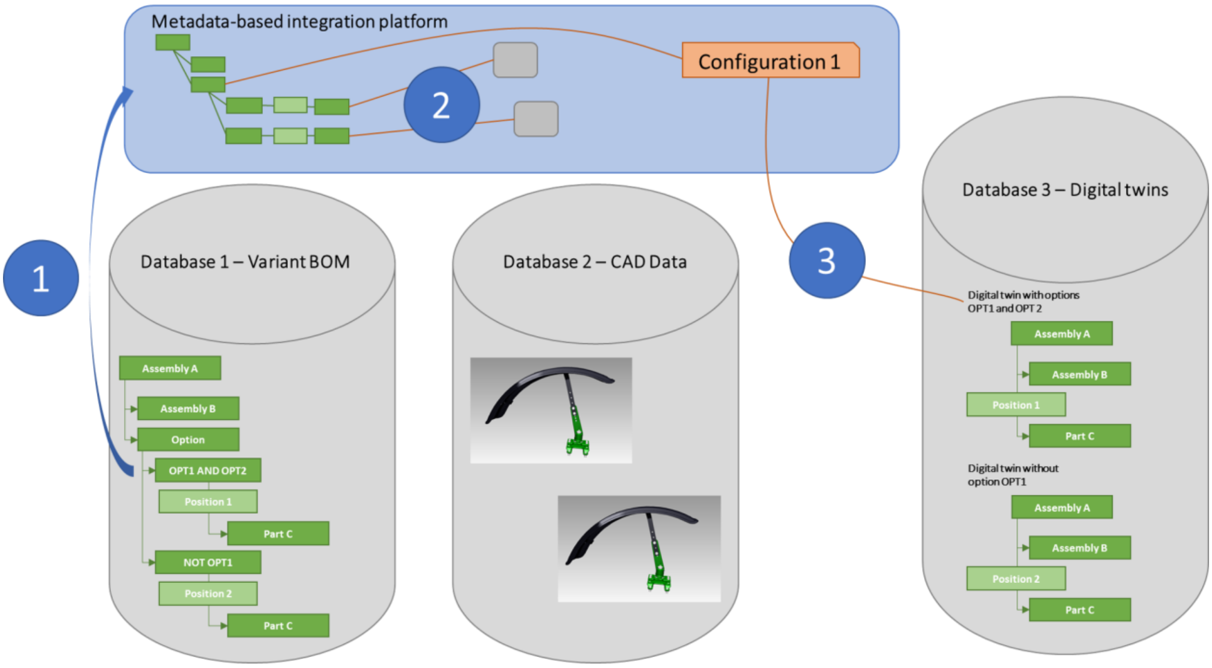


Figure 3. Process for metadata-based creation of digital twins

Different data sets such as different Bills-of-Material (BOMs) can be imported into the metadata graph from their respective data sources, as shown in Figure 3 - (1). For each object, a corresponding metadata graph node is created. These nodes are connected by edges corresponding to relations between data objects in the source system. New edges can be added to link data from different sources together, cf. Figure 3-(2).

In the context of digital twins, this linked metadata can then be used in conjunction with the configuration of a future (or existing) product instance to gather all data that is relevant to this instance. This gathered data can then be reduced to the forms described in the previous section and corresponding data structures can be generated in the database that will store the planned digital twin, depicted in Figure 3-(3).

4. Exemplary Derivation of Digital Twins for Availability-oriented Business Models

The realization of digital product twins in a realistic scenario is a complex task. Even if a specific scope for a future digital twin has been defined in a conceptual design (e.g. by following the framework describe in section 3.1), the actual implementation typically includes the integration of data from

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

different, incompatible sources. To show the viability of the approach presented in this paper, various prototypes can offer an insight into different aspects of the task.

Early stages of the concepts described in section 3 were implemented in the German Federal Ministry of Education and Research (BMBF)-funded research project InnoServPro (Aurich *et al.*, 2019). The project aimed at realizing availability-oriented business models and thus pursued three interrelated sub-goals. The project's sub-goals were the design of a concept for the development of availability-oriented business models and corresponding requirements for their implementation; the technical development of intelligent, communication-capable components; and the development of a communication platform for managing all service-relevant data in the form of a digital twin and making it available via various, role-specific user interfaces.

In the project's three industrial use cases, two manufacturers of agricultural machinery and one supplier of automation technology offered scenarios based around existing products. Instead of selling the products to customers directly, the aim of the research project was to find business models where the customer pays for the product's availability instead (e.g. for the guaranteed availability of a harvesting machine during the harvesting season).

Offering guaranteed availability requires information about the product's configuration and state. For this purpose, a PLM-centric digital twin provided data about critical product components. Based on the expected remaining lifetime, predictive maintenance processes could be started. If components failed, the twin also provided information for repair processes based on the product instance's specific configuration.

In the context of the project, a tightly integrated PLM solution was used. The digital twin was realized as an extension of the PLM data model. Sensor data from the smart products were stored in a separate database, which was embedded into the digital twin's user interface in the form of a dashboard. More specific information about the developed technical solution can be found in (Eickhoff *et al.*, 2020a).

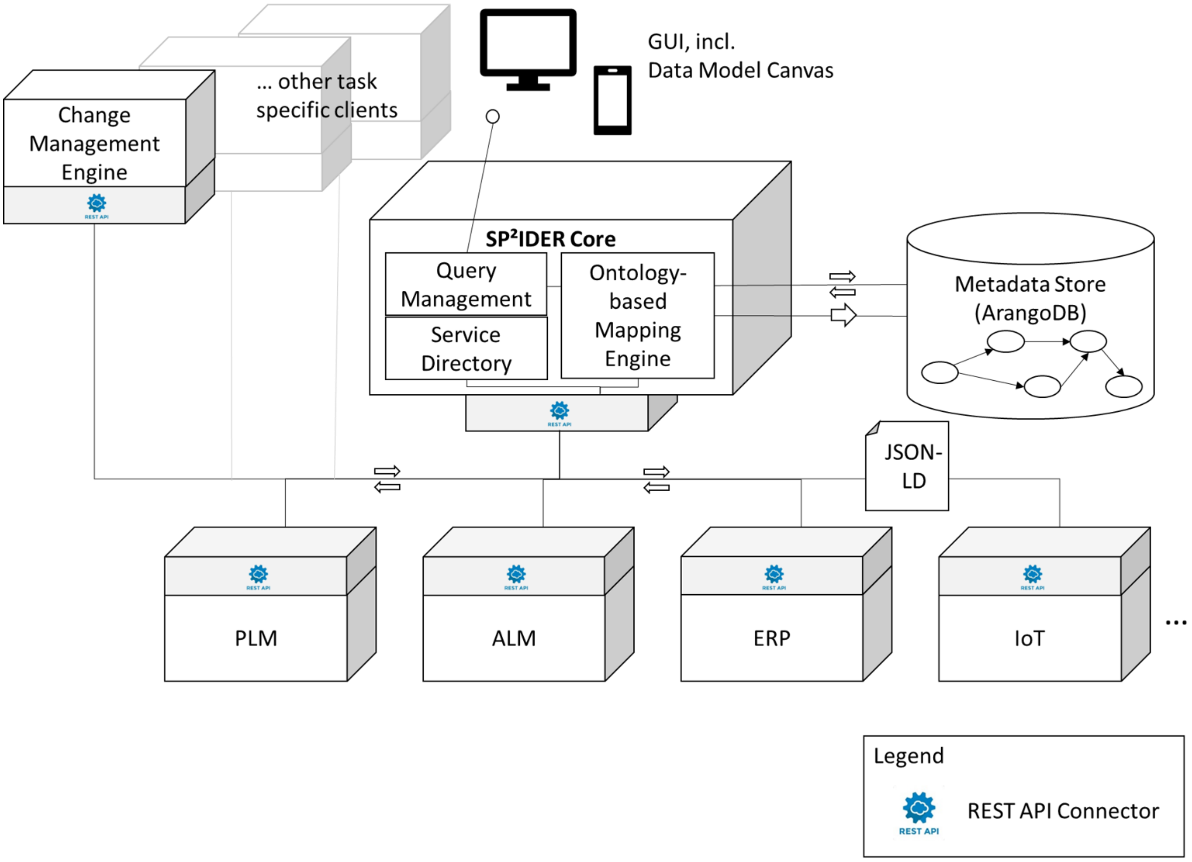


Figure 4. SP^{2}IDER prototype architecture

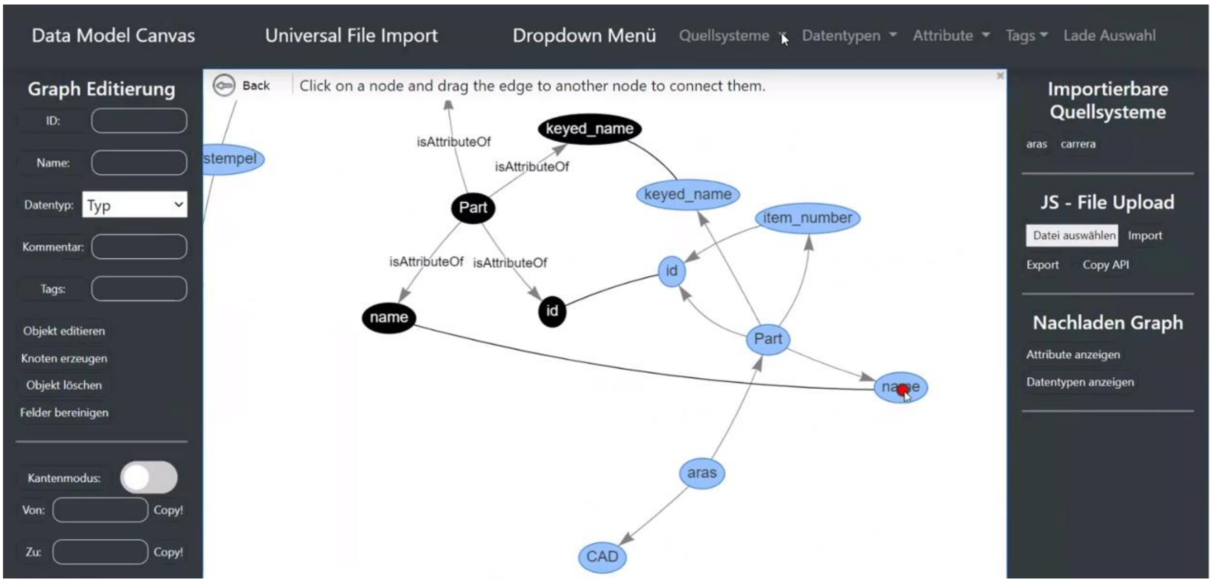
The project's results were then used as the basis for further analysis of the problem space. The prototype developed in the project integrated the digital twin directly into the PLM solution. This is not always

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

practical. Following the ideas described in section 3.3, a platform for the metadata-based integration of different data sources was developed (Eiden *et al.*, 2021; Eickhoff *et al.*, 2020b). The Semantic Product/Process Information and Digitized Engineering Repository (SP^{2}IDER) gathers metadata about the objects present in the source systems (see Figure 4). The metadata is then saved in the graph-based metadata store, where connections across system borders can be introduced as additional edges, forming the basis of a digital thread. The architecture of SP^{2}IDER follows current best practices. Source systems are connected to the core using simple connectors which communicate with the core via RESTful APIs. The data format of these services is built on top of JSON/LD, the Javascript Object Notation for Linked Data as defined in (Kellogg *et al.*, 2019). It is planned to offer some form of compatibility with the OSLC. Realtime data (e.g. from sensors) can be sent using the Message Queuing Telemetry Transport (MQTT) protocol as defined in (Coppen *et al.*, 2019), but is finally stored in one of the source systems and accessed via a JSON-based format as well.

The connected metadata can then be used by several services, which also connect to the SP^{2}IDER core via a RESTful API. The creation of the core's metadata model with interconnections across system borders can be performed by manually adding edges, which is only realistic for small data sets. In addition, the core offers an ontology-based mapping engine, which can add edges based on algorithmic rules or similarities detected by machine learning algorithms. All of these methods require an intuitive user interface, especially when also considering the digital twins as additional sources of data objects.

The Data Model Canvas (DMC) was developed as one such interface (as depicted in Figure 5). The DMC is a user interface for the graph-based analysis and synthesis of data models (Eickhoff *et al.*, 2021). In the context of digital twins, it can help to build the digital thread and to identify anchor points for data from digital twins in the digital model. It is implemented as a service connected to the SP^{2}IDER core described in the previous subsection. The DMC operates by importing data types from one source system and then iteratively integrating new data sources and connecting their data types to the ones previously imported into the graph. Data types are represented as nodes, existing relations between these types are represented as edges. New edges can be added to make implicit connections usable for the system. This is especially useful for connections between data from different source systems. The resulting graph forms a universal data model for the digital thread. This model forms the basis for SP^{2}IDER's metadata graph.



Click on a node and drag the edge to another node to connect them.

Figure 5. Data model canvas prototype

Figure 5 shows an example of the DMC in action. In this scenario, the data types of two source systems are being connected by drawing edges between objects of the first system (black nodes) and the second system (blue nodes). In the example the corresponding data types have the same names in both systems, which is generally not the case. However, if identifying attributes such as similar names or IDs exist, this information can be used to partially automate the generation of the required edges.

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

5. Outlook

This contribution gives a brief introduction to the lifecycle information management in the context of digital product twins. It describes the conceptual design of specific digital twin solutions supported by an appropriate digital twin modelling framework, the Kaiserslautern Digital Twin Modelling Framework (KDTM) and the corresponding metadata-based management of the lifecycle information. As an example for the solutions offered by this approach, an exemplary derivation of digital twins for availability-oriented business models was described. This scenario also highlights the benefits of using a metadata repository like the one offered by the Semantic Product/Process Information and Digitized Engineering Repository (SP²IDER).

The KDTM and the subsequent digital twin derivation process are especially applicable to scenarios where a digital twin can benefit from the connection to existing data sources. In future research, this connection could be further exploited to gain information about product usage and failure states, which could then be used to improve future product generations. The usefulness of this information can be further increased by making it more easily available during typical engineering design workflows. This requires the design of specific tools and assistants which can gather their data from the digital twin and the connected digital model using SP²IDER's service model.

Both SP²IDER's core and the Data Model Canvas are under active development. Our current focus is the different modes of adding digital thread edges to SP²IDER's metadata graph. Additionally, new services that connect to SP²IDER will enable a more effective use of the created graph as well as the connection to the underlying source systems. In particular, it is planned for SP²IDER to enable Access to "real-time" data from digital twins over an MQTT interface.

SP²IDER can also be extended by implementing open standard interfaces to existing solutions. One such "extension" can be achieved by slightly modifying SP²IDER's data format for the communication between the source systems and the core to not only use JSON/LD but also offer an RDF-XML-based format that uses the conventions defined in the OSLC. This could create new synergies between SP²IDER and existing OSLC solutions, as SP²IDER could then become a service provider and offer the capabilities of its services to the OSLC ecosystem.

Even with a robust metadata repository, the introduction of digital twins which are tightly coupled to the digital model offers new challenges and possibilities for existing PLM solutions. In addition to the specific further development of the existing software demonstrators, the management of digital twins themselves can be developed even further. One possible extension is the structured reuse of the data gathered in the digital twin by developing suitable tools and processes for a more full-fledged digital twin lifecycle management. This would allow a tighter integration of digital twins in the development of future product generations.

References

Abramovici, M., Göbel, J.C. and Dang, H.B. (2016), “Semantic data management for the development and continuous reconfiguration of smart products and systems”, *CIRP annals*, Vol. 65 No. 1, pp. 185–188. 10.1016/j.cirp.2016.04.051.

Abramovici, M. and Herzog, O. (2016), Engineering im Umfeld von industrie 4.0: Einschätzungen und Handlungsbedarf, Herbert Utz Verlag.

Alt, O. (2012), *Modellbasierte Systementwicklung mit SysML*, Carl Hanser Verlag GmbH Co KG.

Amsden, J. and Speicher, S. (2021), *OSLC Core Version 3.0. Part 1: Overview*, OASIS. Project Specification Draft. URL: https://docs. oasis-open-projects …

Aurich, J.C., Koch, W., Kölsch, P. and Herder, C. (2019), *Entwicklung datenbasierter Produkt-Service Systeme*, Springer Berlin Heidelberg, Berlin, Heidelberg. 10.1007/978-3-662-59643-2.

Bergsjö, D., Malmqvist, J. and Ström, M. (2006), “Architectures for mechatronic product data integration in PLM systems”.

Bernard, Y. (2012), “Requirements management within a full model-based engineering approach”, *Systems Engineering*, Vol. 15 No. 2, pp. 119–139. 10.1002/sys.20198.

Coppen, R., Banks, A., Briggs, E., Borgendale, K. and Gupta, R. (2019), *MQTT Version 5.0*, available at: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2006), “Ontological Engineering: Principles, Methods, Tools and Languages”, in *Ontologies for software engineering and software technology*, Springer, pp. 1–48. 10.1007/3-540-34518-3_1.

Eickhoff, T., Apostolov, C. and Göbel, J.C. (2020a), “Methodologically supported development of digital twins for smart product-service systems”, *Digital Proceedings of TMCE* 2020, pp. 155–164.

Eickhoff, T., Eiden, A., Göbel, J.C. and Eigner, M. (2020b), “A Metadata Repository for Semantic Product Lifecycle Management”, *Procedia CIRP*, Vol. 91, pp. 249–254. 10.1016/j.procir.2019.11.006.

Eickhoff, T., Eiden, A., Gries, J. and Göbel, J.C. (2021), “Data Model Canvas für die IT-Systemübergreifende Integration von Datenmodellen zur Unterstützung von Datenanalyse-Anwendungen im Produktlebenszyklus”, pp. 99–109. 10.25368/2021.14.

Eiden, A., Eickhoff, T., Gries, J., Göbel, J.C. and Psota, T. (2021), “Supporting semantic PLM by using a lightweight engineering metadata mapping engine”, *Procedia CIRP*, Vol. 100, pp. 690–695. 10.1016/j.procir.2021.05.146.

Forte, S., Göbel, J.C. and Dickopf, T. (2021), “SYSTEM OF SYSTEMS LIFECYCLE ENGINEERING APPROACH INTEGRATING SMART PRODUCT AND SERVICE ECOSYSTEMS”, *Proceedings of the Design Society*, Vol. 1, pp. 2911–2920. 10.1017/pds.2021.552.

Friedenthal, S., Moore, A. and Steiner, R. (2014), *A practical guide to SysML: the systems modeling language*, Morgan Kaufmann.

Göbel, J.C. and Eickhoff, T. (2020), “Konzeption von Digitalen Zwillingen smarter Produkte”, *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Vol. 115 No. s1, pp. 74–77.

Hildebrandt, C., Torsleff, S., Caesar, B. and Fay, A. (2018), “Ontology Building for Cyber-Physical Systems: A domain expert-centric approach”. 10.1109/COASE.2018.8560465.

ISO 15288 (2015), Systems and software engineering — System life cycle processes No. 15288:2015.

Kellogg, G., Champin, P.-A. and Longley, D. (2019), “JSON-LD 1.1-A JSON-based Serialization for Linked Data”, W3C, 2019.

Kuhn, T. (2017), “Digitaler Zwilling”, *Informatik-Spektrum*, Vol. 40 No. 5, pp. 440–444. 10.1007/s00287-017-1061-2.

Lentes, J., Eckstein, H. and Zimmermann, N. (2012), “A Platform to Integrate Manufacturing Engineering and Product Lifecycle Management”, *IFAC Proceedings Volumes*, Vol. 45 No. 6, pp. 1071–1076. 10.3182/20120523-3-RO-2023.00425.

Porter, M.E. and Heppelmann, J.E. (2014), “How smart, connected products are transforming competition”, *Harvard business review*, Vol. 92 No. 11, pp. 64–88.

Savarino, P. (2019), Dynamische Ermittlung echtzeitkompatibler internetbasierter Services für smarte kundenindividuelle Massenprodukte in deren Nutzungsphase, Shaker Verlag.

Schroeder, G.N., Steinmetz, C., Pereira, C.E. and Espindola, D.B. (2016), “Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange”, *IFAC-PapersOnLine*, Vol. 49 No. 30, pp. 12–17. 10.1016/j.ifacol.2016.11.115.

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J. and Wang, L. (2012), “Modeling, simulation, information technology & processing roadmap”, *National Aeronautics and Space Administration*, Vol. 32, pp. 1–38.

Stark, R., Kind, S. and Neumeyer, S. (2017), “Innovations in digital modelling for next generation manufacturing system design”, *CIRP annals*, Vol. 66 No. 1, pp. 169–172. 10.1016/j.cirp.2017.04.045.

Zeimetz, T. and Schenkel, R. (2020), “Sample Driven Data Mapping for Linked Data and Web APIs”. 10.1145/3340531.3417438.

https://doi.org/10.1017/pds.2022.158 Published online by Cambridge University Press