[PZC+25] J. Pfeiffer, J. Zhang, B. Combemale, J. Michael, B. Rumpe, M. Wimmer, A. Wortmann:
Towards a Unifying Reference Model for Digital Twins of Cyber-Physical Systems.
In: 2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1-4,
DOI 10.1109/ETFA65518.2025.11205646, Sep. 2025.

Towards a Unifying Reference Model for Digital Twins of Cyber-Physical Systems

Jérôme Pfeiffer
University of Stuttgart
Stuttgart, Germany
jerome.pfeiffer@isw.uni-stuttgart.de

Jingxi Zhang
University of Stuttgart
Stuttgart, Germany
jingxi.zhang@isw.uni-stuttgart.de

Benoit Combemale
University of Rennes
Rennes, France
benoit.combemale@irisa.fr

Judith Michael
RWTH Aachen University
Aachen, Germany
michael@se-rwth.de

Bernhard Rumpe
RWTH Aachen University
Aachen, Germany
rumpe@se-rwth.de

Manuel Wimmer
JKU Linz
Linz, Austria
manuel.wimmer@jku.at

Andreas Wortmann
University of Stuttgart
Stuttgart, Germany
wortmann@isw.uni-stuttgart.de

***Abstract*—Digital twins are sophisticated software systems for the representation, monitoring, and control of cyber-physical systems, including automotive, avionics, smart manufacturing, and many more. Existing definitions and reference models of digital twins are overly abstract, impeding their comprehensive understanding and implementation guidance. Consequently, a significant gap emerges between abstract concepts and their industrial implementations. We analyze popular reference models for digital twins and combine these into a significantly detailed unifying reference model for digital twins that reduces the concept-implementation gap to facilitate their engineering in industrial practice. This enhances the understanding of the concepts of digital twins and their relationships and guides developers to implement digital twins effectively.**

***Index Terms*—Digital Twin, Industry 4.0, Reference Model**

I. Introduction

Research and industry employ digital twins (DTs) [1], [2] for various kinds of cyber-physical systems (CPSs), in many domains, including automotive, construction, energy management, medicine, smart manufacturing, and more [3], [4]. They recently have been standardized in manufacturing [5] to become a central part of automated manufacturing systems. Yet, despite various definitions, there is little consensus about what a DT actually is and what it should comprise. To mitigate this, research and practice have produced various reference models of DTs, which aim to explain their main constituents and relations. Advancing the engineering of DTs requires establishing a detailed and comprehensive grasp of their constituents and their relationships. To guide their systematic engineering, we present a unifying reference model of DTs of greater detail that helps reduce the concept-implementation gap by introducing consistent descriptions of common abstract concepts proposed in the existing reference models. Our contributions to the engineering of DTs, hence, are (1) an analysis of reference models of DTs from different domains, that we selected based one popularity, and (2) a synthesis of a detailed reference model for DTs that reduces the concept-implementation gap.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Model-Based DevOps – 505496753. – Funded by the Agence Nationale De La Recherche (ANR) – France – Website: https://mbdo.github.io.

II. Reference Models of Digital Twins

A. Reference Models Focusing on Data Flows

A very popular characterization of DTs [1], which subsumes the definition by Grieves [6], distinguishes them from digital models and digital shadows according to the data flows between the actual system (AS), *i.e.,* the CPS, and the digital object [1]: (1) *Digital model*: If there are no automated data flows between the AS and the digital object, the digital object is considered a digital model, such as a CAD model of a car. (2) *Digital shadow*: If there is an automated data flow from the AS to the digital object, the digital object is called a digital shadow (in the sense that the digital object automatically follows certain changes in the physical object), such as the representation of various properties of interest, *e.g.,* mileage, speed, on a modern car's digital dashboard. (3) *Digital twin*: If there is an automated data flow back from the digital object to the AS as well, the digital object is considered a DT, as changes on the digital object then also entail changes on the AS. And while these categories enable a qualitative distinction between digital models, digital shadows, and DTs, they are at a high level of abstraction as they omit to specify any details on the constituents of DTs.

B. Reference Models Focusing on Components

The 5D model of DTs [2] proposes that DTs consist of five kinds of elements: (1) *Physical entities* exist within the physical world and serve as the underlying basis for DTs. (2) *Virtual models* replicate physical entities, as well as their physical properties and behaviors. (3) *Digital twin data* stems from the physical entity, services, domain expert knowledge, or a combination of different sources. (4) *Services* provided by DTs offer added-value functions, e.g., simulation, verification, monitoring, optimization, diagnosis, prediction. (5) *Connections* in the DT connect the four components to enable collaboration between them.

TABLE I
COMPARISON OF THE MAIN CONCEPTS WITHIN THE DISCUSSED REFERENCE MODELS OF DTS.

Main Concept	Kritzinger et al. [1]	Tao et al. [2]	ISO 23247 [5]	IDTA (AAS) [7]
Physical Object	physical object	physical entity (PE)	observable manufacturing element	asset
Digital Object	digital object	virtual models	digital representation (DR)	digital representation from a given viewpoint (AAS)
Model	digital model	virtual entity (VE, a set of models)	(subsumed by DR)	submodel
Data	data flows	digital twin data	data collection entity	data element
Device Communication	data flows	connection	device communication	manifest, AAS interfaces
Service	-	services (of PE and VE)	services as part of DT entity	service
Digital Twin	digital twin	digital twin	digital twin	digital twin

C. Reference Models from Industry

1) ISO 23247: This standard describes a reference model of a DT framework for manufacturing [5]. The functional view of this model comprises three layers of entities that provide functionality over observable manufacturing elements (OMEs). OMEs are items providing observable properties (*e.g.,* a manufacturing plant, a service robot, or staff), and their properties are defined through existing manufacturing standards. The *device communication entity layer* comprises functional elements (FEs) to collect and process data from OMEs, as well as to control and actuate OMEs. The *DT entity layer* uses the device communication entity to, among other functions, represent, manage, operate, simulate, and maintain the devices observed through the OMEs. Through the *user interface entity layer*, users and additional services can leverage the DT entity and its potentially built-in service sub-entity to reason about and control the twinned system.

2) Asset Administration Shell: The Industrial Digital Twin Association (IDTA) is developing the Asset Administration Shell (AAS), a standard and framework to collect all information (models, data, documents) of an asset, *e.g.,* a milling machine or a robot, systematically and in a machine-processable fashion. The IDTA envisions three kinds of AAS [8]: the type 1 AAS consists of documents and models; the type 2 AAS additionally collects operation data from the asset it is connected to (e.g., via OPC UA); and the type 3 AAS has built-in analytics to reason about the asset and control it. The type 3 AAS, hence, can be understood as an implementation of the DT definition proposed by Kritzinger et al [1].

D. Comparing the Concepts

We compare the concepts as outlined in Table I. As the reference model of the IDTA [7] includes concepts from DIN Spec 91345 [9], we have combined these within one column.

1) Physical Object (according to [1], [2]): Current approaches for DTs show that the actual system can be anything, from a cyber-physical to a socio-technical, biological, or organizational system. Clearly, any physical system itself cannot be part of a *digital* twin. Instead, the interfaces and connections between the DT and the AS need to be detailed.

Conclusion C1: A DT cannot contain its physical twin but needs interfaces to obtain data from and send commands to it.

2) Digital Object (according to [1]): Opposed to the physical object, the literature refers to its digital representation as, *e.g., digital object* [1], *virtual models* [2], *digital representation* [5], or *digital representation from a given viewpoint* [7]. Additionally, the IDTA states that the AAS is the "standardized digital representation of an asset" [7], where an asset is anything of value. Harmonizing these terms within a reference model requires further concretization of what such a digital object can be. Furthermore, the different conceptualizations show that a DT has to include both data as well as models, such that it can be called a DT.

Conclusion C2: Digital objects can be on different abstraction levels ranging from specifically recorded data to general models used for different purposes as is discussed next.

3) Model (according to [1], [2]): Kritzinger et al. use the term *digital model* [1] to describe the digital representation of an existing or planned object. Tao et al. describe the *virtual entity* as a set of *models* (geometry model, physics model, behavior model, and rule model) [2]. ISO 23247 uses the term *representation* to describe the "manner in which information is modeled for interpretation by a machine" [5]. The IDTA uses the term *submodel* as a "container of SubmodelElements defining a hierarchical structure consisting of SubmodelElements" [7]. Obviously, leveraging models requires the software infrastructure to manage (create, update, delete, search, relate) them among each other and with the data recorded by the DT, *e.g.,* to synchronize observed property changes in the twinned AS with the relevant models the DT has about these properties (and vice versa).

Conclusion C3: A DT requires various kinds of models. Hence it needs to manage them, and synchronize properties with properties about the twinned system.

4) Data (according to [1], [2], [5], [7]): Kritzinger et al. have a specific focus on *data* and the properties of

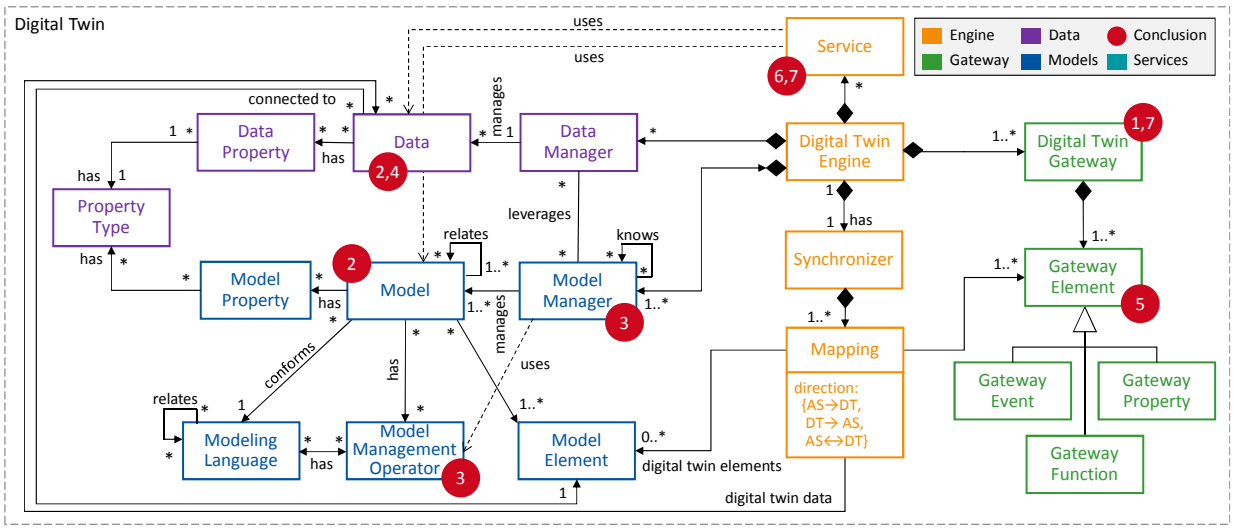

Fig. 1. The main DT concepts are a gateway connecting to the AS, models and data representing it, an engine synchronizing properties of interest between both, and added-value services.

the data flow, i.e., manual or automatic [1]. Tao et al. use the term *DT data*, which they denote based on its source (the physical, virtual entity, services, domain knowledge and obtained through data fusion) [2]. ISO 23247 uses the term *data* [5] with a specific focus on its requirements. i.e., data acquisition, analysis, and integrity. The IDTA defines data elements, where "a data element is a submodel element that is not further composed of other submodel elements and has a value. This means that DTs will ingest, process, manage, and output various kinds of data (structured data, streams, configurations, ...) and need the infrastructure to persist it, perform computations on it, attach metadata, and generally, manage it. To make sense of the data, the DT needs to be able to relate data to models.

Conclusion C4: A DT needs to manage different kinds of data and related metadata as well as relate data to its models.

5) Device Communication (according to [5]): Tao et al. define bi-directional *connections* as a main concept between the different DT constituents [2]. ISO 23247 refers to *device communication* as a relevant concept between a DT and the observable manufacturing element [5], which entails data collection and device control. The IDTA [7] refers to the DIN spec 91345 mentioning the term *manifest* as "externally accessible, defined set of meta information that provides information about the functional and non-functional properties of the I4.0 component" [9] and DIN EN IEC 63278-1 refers to the *asset administration shell interfaces* as communication provider: Hence, the communication between the DT and the AS is bidirectional, i.e., the DT can send control commands to the AS.

Conclusion C5: A DT needs to read data from and send a command to its AS. To be able to react upon changes in the AS, it also needs to be able to observe changes in the AS.

6) Service (according to [5], [7], [9], [10]): Tao et al. use the term *services* for the physical entity and the virtual entity [5]. ISO 23247 uses the term *services* as functions provided by the DT entity [5]. The IDTA uses the term *service* with the meaning of a "demarcated scope of functionality which is offered by an entity or organization via interfaces" (from [7] as an extension of [9]). Overall, the commonality of the different interpretations of the term service is that everything not vital to the DT's core functionality of representing the AS digitally can be considered a service (including analytics, AI, etc.).

Conclusion C6: A DT needs means to interface added-value services on top of its core functionality.

7) Digital twin (according to all reference models): All contributions use the term *digital twin* with slightly different definitions, but most do not explicitly specify where the border of a DT is, *e.g.,* [1], [2], [7].

Conclusion C7: A DT needs well-defined boundaries to its environment including the AS and its services.

III. Unifying Digital Twin Reference Model

Our reference model for DTs (*cf.* Fig. 1) is based on the outlined conclusions.

Digital Twin Engine. At the heart of the DT is the `Digital Twin Engine` (*cf.* Fig. 1), which connects the gateways, the models, the data and the services of the DT. It

comprises a Synchronizer over Mappings between the Model Elements managed by the DT and the Gateway Elements representing the ASs (C3). A Mapping specifies a direction, which can be unidirectional, from AS to DT or vice versa, or bidirectional (C4). Moreover, a Mapping can be either synchronized at a specified frequency or at the occurrence of a specific trigger. Overall, the Digital Twin Engine connects the Gateways to Data Managers, Model Managers, and eventually Services.

Digital Twin Gateway. To gather data from the twinned ASs directly or from systems observing the twinned ASs, each DT connects to one or more Gateways (C4). Each Gateway represents a AS and exposes AS features of interest as Gateway Elements to the DT engine (C1). These features either are (1) Properties that can be read, observed, and synchronized. (2) Events that the DT engine and the AS may react to. (3) Functions that can be invoked on the Gateway to manipulate the twinned AS, which can receive arguments and return results, as known from programming.

Data. Data has to be collected by the Digital Twin Engine through the Data Manager that ensures well-formed storage access to specific databases or similar systems (C2). Data are associated with some Data Properties qualifying it. Those properties are of Property Types such as the time (live, historical), if it is raw data or processed one, the origin of the data (e.g., the AS, or DT services), the uncertainty, the precision, the last update, etc.

Digital Twin Models. If the Digital Twin Engine is the heart of the DT, the Models are its brain, which may only be changed through the Model Manager or the synchronization of related Mappings. Each model conforms to a Modeling Language and comprises Model Properties of interest, which can belong to different Property types, out of which the latest update is mandatory for DTs supporting bidirectional synchronization (C3). The languages and the models themselves can have Model Management Operators that govern how the models can be changed while retaining their intra- and inter-model integrity (C3). The DT engine accesses and manipulates its models only through these Model Managers that are responsible for certain models and embody the language engineering expertise to safely manipulate these. To this end, each Model Manager can delegate changes to other model managers. Moreover, the models can be used either offline, *i.e.,* not connected to a Gateway, *e.g.,* to preserve a certain state of the model or to use it for experimentation without affecting the ASs, or online, *i.e.,* connected to a Gateway, such that changes to the model can affect an AS and vice-versa (C3) [2]. Such information is captured as a Model Property.

Digital Twin Services. Finally, the Digital Twin Engine can purposefully leverage its models and data to provide added-value functionalities through services, such as monitoring key performance indicators, predicting maintenance, or representing the behavior of the twinned system in a specific fashion. Therefore, Services can interact with the Digital Twin Engine (C5) and, through it, may interact with the DT Gateways. However, this interaction may be restricted by the Digital Twin Engine (C6) to prevent undesired changes to the DT's models (C3) and the AS.

IV. Outlook

We analyzed and compared selected reference models from research, industry, and standardization associations. Based on these, we devised an initial unifying model that captures the essential concepts of the analyzed models and refines these by detailing some parts of it. To further detail our reference model, we will investigate (a) further conceptual models of DTs, such as [10]–[12], (b) analyze reference models of services, (c) refine the connection between models and data in our model, and consider the conceptual models implemented by DT platforms [13]. Then we plan to elaborate on how this unifying model can be translated into DT implementations and underpin this with technological options for each part. Both will support researchers and practitioners in analyzing and engineering DTs.

References

[1] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline*, vol. 51, no. 11, pp. 1016–1022, 2018.

[2] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[3] M. Dalibor, N. Jansen, B. Rumpe, D. Schmalzing, L. Wachtmeister, M. Wimmer, and A. Wortmann, "A cross-domain systematic mapping study on software engineering for digital twins," *Journal of Systems and Software*, p. 111361, 2022.

[4] J. Michael, J. Blankenbach, J. Derksen, B. Finklenburg, R. Fuentes, T. Gries, S. Hendiani, S. Herlé, S. Hesseler, M. Kimm, J. C. Kirchhof, B. Rumpe, H. Schüttrumpf, and G. Walther, "Integrating Models of Civil Structures in Digital Twins: State-of-the-Art and Challenges," *Journal of Infrastructure Intelligence and Resilience*, vol. 3, no. 3, 2024.

[5] ISO, "ISO 23247 automation systems and integration-digital twin framework for manufacturing," tech. rep., International Standardization Organization (ISO), 2020.

[6] M. W. Grieves, "Product lifecycle management: the new paradigm for enterprises," *International Journal of Product Development*, vol. 2, no. 1-2, pp. 71–84, 2005.

[7] Industrial Digital Twin Association, "Specification of the Asset Administration Shell. Part 1: Metamodel," 2023.

[8] J. Zhang, C. Ellwein, M. Heithoff, J. Michael, and A. Wortmann, "Digital Twin and the Asset Administration Shell: An Analysis of 3 AASs Types and their Feasibility for Digital Twin Engineering," *Journal Software and Systems Modeling (SoSyM)*, 2025.

[9] "DIN SPEC 91345:2016 Reference Architecture Model Industrie 4.0 (RAMI4.0)," 2016.

[10] Digital Twin Consortium, "Platform Stack Architectural Framework: An Introductory Guide," 2023.

[11] R. Eramo, F. Bordeleau, B. Combemale, M. van Den Brand, M. Wimmer, and A. Wortmann, "Conceptualizing Digital Twins," *IEEE Software*, vol. 39, no. 2, pp. 39–46, 2021.

[12] J. Pfeiffer, D. Lehner, A. Wortmann, and M. Wimmer, "Towards a product line architecture for digital twins," in *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*, pp. 187–190, IEEE, 2023.

[13] J. Pfeiffer, D. Lehner, A. Wortmann, and M. Wimmer, "Modeling capabilities of digital twin platforms - old wine in new bottles?," *J. Object Technol.*, vol. 21, no. 3, pp. 3:1–14, 2023.