Digital Twin Core Conceptual Models and Services

Essential Models for Interoperability

An Industry IoT Consortium Best Practices Paper

2023-11-02

Authors

Shi-Wan Lin (Yo-I Information Technologies, Co. Ltd), Kym Watson (Fraunhofer IOSB), Guodong Shao (NIST), Ljiljana Stojanovic (Fraunhofer IOSB), Bassam Zarkout (IGnPower Inc.)

CONTENTS

FIGURES

TABLES

1 Overview

Enterprises are digitally transforming themselves to improve operational efficiency, enhance competitiveness and create new value[1]. Digital transformation often involves creating synchronized virtual representations of real-world entities, widely referred to as *digital twins*.

Digital twins provide a systematic way for operators and others to represent, model, analyze and optimize systems and processes throughout their lifecycle from planning, design, modeling and simulation, validation and build, deployment, operation and maintenance, and decommissioning. Digital twins enable engineers to test design changes or perform post-analyses on operational events without costly and high-risk experiments with real components. Based on real-time and historical data, digital twins enhance operators' understanding of current system conditions, predict future trends and make informed decisions.

Digital twins support feedback loop(s) in operation based on data involving sensing, analyzing, decision-making and acting, possibly synchronized. The feedback loop may involve (and increasingly without) humans in the loop. Key to this feedback loop is operational decision-making enabled by insights from analytics and simulations using data reflecting the dynamic states of the physical entities and/or processes. This benefits product design, process engineering, production process management, quality management, energy management and asset maintenance, especially when the real-world processes are complex and dynamic. This data-driven decision-making capability enables engineers and operators to predict and validate designs, and detect and mitigate, or predict and prevent, exceptions in operation processes.

Digital twins must be highly interoperable, both with the real-world entities and other software systems (including other digital twins) running in the operational environment that are often supported by tools from different vendors. Standards-based approaches support interoperability and encourage the reuse of components.

Predefined standard-based digital twins of equipment, subsystems and processes enable "plug-and-play" through standard interfaces to support different business applications in various operational environments. To enable such "plug-and-play," standard digital twin interfaces need to be developed and utilized. Standards-based interoperable approaches enable digital twins and related systems to be connected, synchronized, and trusted by each other.

Existing applicable standards may be used to achieve only parts of the interoperability goal since the concept and the applications of digital twins are still in their development stage. At this stage,

[1] *https://www.iiconsortium.org/pdf/Digital_Transformation_in_Industry_Whitepaper_2020-07-23.pdf*

there are active developments of standards to meet the comprehensive needs of digital twins by many organizations. The applications of digital twins are broad in many usage scenarios spanning many diverse industries. Though sharing general commonalities with each other, each of these applications may also possess unique requirements in specific environments. Therefore, approaches to address these diverse requirements require careful examination and exploration from various perspectives so they are sufficiently flexible and highly adaptable.

This report introduces Digital Twin Core Conceptual Models and Services, that seamlessly connect the foundational IT infrastructure (the lower layer) and industry-specific business applications (the upper layer) in industrial settings empowered by digital twins (see Figure 1-1). By positioning the "digital twin core" as a middle layer that encapsulates the key digital twin functionality, it makes it easier to focus on digital twin capabilities and their standardization. Moreover, it offers increased adaptability in the deployment of the IT infrastructure and specialized business applications, both of which exhibit immense diversity and a wide array of implementation options. It also reduces the scope and the challenges in the standardization of digital twins.

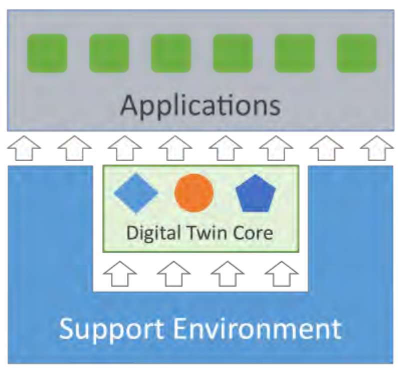


Figure 1-1: Digital twin core model.

This paper describes the fundamentals and basic requirements of digital twins, the digital twin core conceptual models and services, the enabling architectures and technologies, and supported business applications.[2] It provides certain high-level technical considerations in the implementation of digital twin core layer and illustrates potential new standardization opportunities for digital twins. As such, the core conceptual models and services described in this report are not standard specifications but representative of ideas to serve as inputs for standards development.

[2] For general information of digital twin's applications in the industries, refer to the IIC whitepaper, "Digital Twins for Industrial Applications"
https://www.iiconsortium.org/pdf/IIC_Digital_Twins_Industrial_Apps_White_Paper_2020-02-18.pdf;
https://www.iiconsortium.org/pdf/Digital-Twin-and-Asset-Administration-Shell-Concepts-and-Application-Joint-Whitepaper.pdf;
https://doi.org/10.3390/automation303002

The target audience of this report includes technical decision makers, system engineers, software architects, modelers and standards development organizations (SDO), who are interested in, involved in and responsible for designing and implementing systems supported by interoperable digital twins.

Recently the Digital Twin Consortium (DTC) published a technical brief entitled "Platform Stack Architectural Framework: An Introductory Guide"[3]. This technical brief 1) clarified digital twin system central concepts and capabilities; 2) reviewed several commonly adopted technological approaches and standards, and, most noticeably; 3) introduced an architectural framework of key common elements of digital twin systems.

The Platform Stack Architecture Framework introduced by DTC lays the groundwork for constructing digital twin systems that are both composable (built by a combination of capabilities) and federated (across organizational lines). This strategy of developing composable digital twin systems offers an efficient way to handle the intricacy of such systems. In discussing composable or federated digital twins, the DTC technical brief considers integration service interfaces and synchronization methods, in addition to security, trust, and governance across the system boundaries of digital twins.

The Digital Twin Core Conceptual Models and Services described by this technical report focuses on defining core conceptual models and services of digital twin systems in a way to maximize the interoperability of digital twin implementations. Notably, the digital twin core layer can be viewed as being contained in the Virtual Representation layer of the Platform Stack Architecture Framework, as shown in Figure 1-2. Moreover, the Support Environment and Applications layers bear one-to-one correspondence to the IT/OT Platform and the Applications components in the Platform Stack Architecture Framework. The Digital Twin Core Conceptual Models and Services technical report underlines the critical interplay of various components and creates a basis for an integrated understanding of digital twin systems.

[3] *https://www.digitaltwinconsortium.org/platform-stack-architectural-fram-formework-an-introductory-guide-form/*

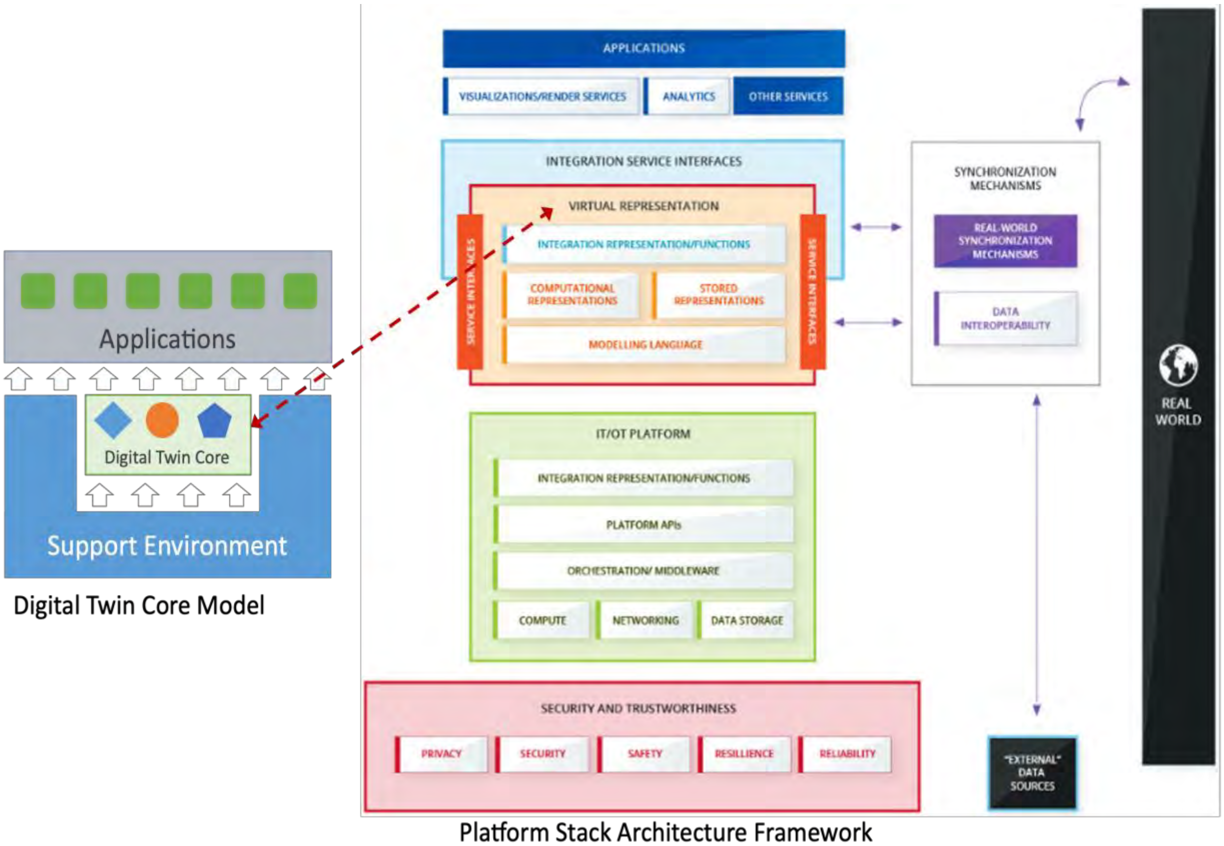


Figure 1-2: Relation between IIC Digital Twin Core Model to the DTC Platform Stack Architecture Framework.

Overall, the Digital Twin Core Conceptual Models and Services Technical Report and the Platform Stack Architectural Framework provide complementary ideas by examining digital twins from different perspectives and together can benefit readers who are building digital twin systems in their diverse applications across industries.

Another important concept related to digital twins is a system of systems (SoS). Digital twin systems, just like their real-world counterparts, are increasingly interconnected, therefore becoming ever more complex and dynamic in their composition, relations, and governance. Many of the digital twin systems resemble or fit into the models of SoS. Consequently, the principles and approaches in SoS may be applicable and beneficial when designing and building digital twin systems.

A joint working group by IIC and DTC is currently working on a separate paper, to be published in the near future, to further explore digital twin composition and federation based on the SoS concept and models to 1) support value creation from interconnected digital twin systems; 2) make existing system capabilities more accessible; and 3) foster emergent new capabilities. This SoS paper focuses on the key benefits of an SoS approach especially in the digital twin context, with its dynamism in forming larger systems and emergent capabilities. It illustrates the value generated from these dynamically formed capabilities in several cross-industry use cases that benefit from building ever larger and increasingly interconnected digital twin systems.

This report focuses on the Digital Twin Core Conceptual Models and Services and does not delve into aspects of SoS in digital twin systems.

Finally, this report includes two appendices, "the Context of Digital Twin" and "Digital Twin as a Paradigm and Enabling Technology," as background material to help the readers better understand some fundamental concepts and ideas that the main sections are based upon.

2 Digital Twin Definition

The term digital twin is used widely in varying contexts and its definition varies as well. A general definition is provided in the glossary of the Digital Twin Consortium and subsequently adopted in the Industry IoT Consortium (IIC) Vocabulary:

> *A digital twin is a virtual representation of real-world entities and processes, synchronized at a specified frequency and fidelity* [4]*.*

This report uses this as its working definition.

The purpose of digital twins is to represent the characteristics and behaviors of real-world counterparts in the digital world, to model and analyze their states, conditions and behaviors so to enhance our understanding of and optimize the operation of these real-world assets.

The focus of this report is the middle layer for building digital-twin-enabled industrial applications. The digital twin core (henceforth, simply 'core') is where the essential digital-twin functions and services are abstracted, encapsulated and provided. The objective is to implement the core as a set of standardized software components that can be implemented with maximum commonality and reusability. This enables interoperability and facilitates cost-effective digital twin system implementations in aiding digital transformation across industrial environments.

[4] The term 'real-world' entity used in this document encompasses physical entities such as a component or a piece of equipment, and logical entities such as a manufacturing process, a product organization or teams. Both are the targets can be represented by digital twins.

3 CONCEPTUAL MODEL OF CORE ELEMENTS

Digital twin is a paradigm and a set of technologies for building a digital representation of entities in the real world so we can understand and operate them better.[5]

Digital twin comprises a set of technologies to represent, in the digital world, across the lifecycle of real-world entities, predominantly equipment and products in industries. From these digital representations, we can systematically and dynamically

- Gather data from real-world entities such as equipment and industrial automation and control systems or from sensors attached to them,
- Perform analytics, simulation and optimization over these data in accordance with the specific properties[6] of the corresponding real-world entities in specific application contexts to garner insights about their operational states or status, and
- Apply these insights to make optimal decisions dynamically in industrial operations.

3.1 DIGITAL TWIN FEEDBACK LOOPS

By forming dynamic and (near-) real-time bidirectional links with their real-world counterparts, digital twins can facilitate closed feedback loops, either completely automated or involving human-in-the-loop, from data (sensing) to insights (analytics), from insights to decisions and from decisions to actions, see Figure 3-1.

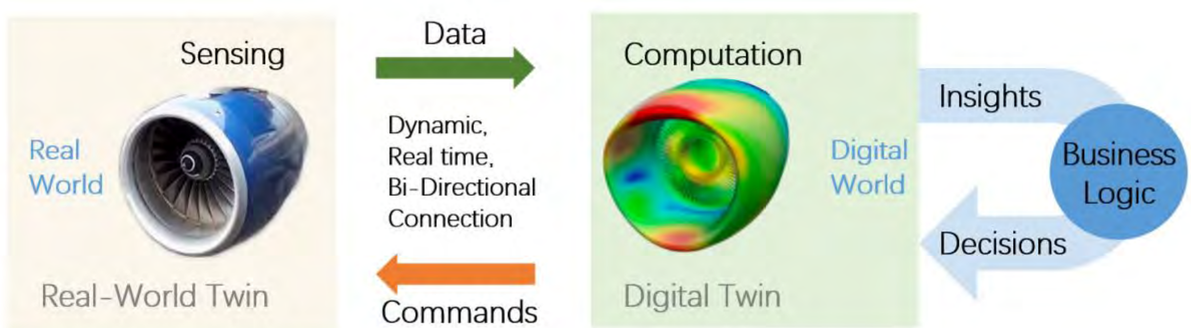


Figure 3-1: Digital and real-world twins.

Through this loop, digital twins can reduce equipment down-time, improve production processes, enhance product quality, increase production efficiency, reduce material and energy cost and strengthen production safety. Digital twins can also be used in broader scenarios that

[5] For more details about digital twin as a paradigm and a set of technologies, refer to Appendix B.

[6] We use property and data generally with meanings consistent with their respective common usages. At a technical level, we use property both as characteristics, e.g. the hardness of the surface and metadata such as that defined in Dublin Core (ISO 15836-1:2017 Information and documentation—The Dublin Core metadata element set—Part 1: Core elements). Data are the actual values of the measurements of properties. Defining a property binds semantic meaning to data associated with it.

may not form direct closed feedback-loops, such as product design and production process engineering.

3.2 DIGITAL TWIN DATA AND COMPUTATIONAL MODELS

The data models and computation models for analytics and simulations, as illustrated in Figure 3-2, are needed to represent the real-time states and behaviors of their real-world counterparts.[7]

Building the data model and computation models for digital twins requires deep domain expertise with comprehensive understanding of the intrinsic properties of the specific real-world entities. These properties include their working principles, design specifications, operational properties and procedures. The development of computation models often requires a significant amount of quality operational data to leverage advanced data-oriented computational algorithms such as those based on ML. Moreover, there could be different implementations of digital twins corresponding to the same real-world entities, each addressing a specific problem domain with unique computation models.

Various computation models are needed for performing analytics or simulations in digital twins to identify anomalies and capture exceptions, diagnose their causes and predict trends about their corresponding real-world entities. Traditionally, physical law and engineering-principle-based methods are used; now advanced analytics, machine learning (ML) and artificial intelligence (AI) are playing an increasingly important role.

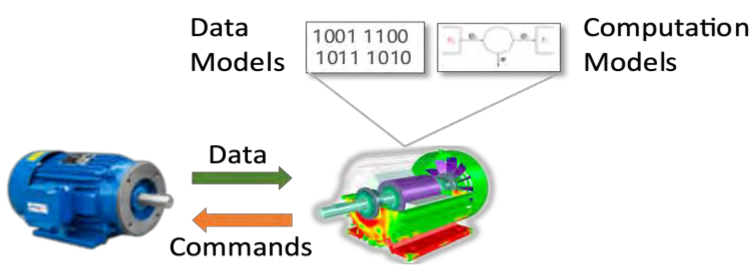


Figure 3-2: Data and computation models in a digital twin.

Combining data models and computation models together as digital twins to represent the properties and behaviors of real-world entities provides a systematic approach to model the real world to solve complex problems. Though requiring deep domain knowledge to build and a stringent process to validate, such digital twins[8] once established can be reused across various scenarios and thus reduce the effort and cost required in broad applications.

---

[7] We use data model as a general term making no distinctions between it and information model.

[8] When we discuss digital twins in the sense of implementation, we loosely refer to them as a set of externalizable models, including configurations and codes, that can be plugged into a certain environment to be realized and executed in software runtime environments.

3.3 DIGITAL TWIN INTEROPERABILITY

To facilitate a new generation of industrial applications in operational environments powered by the digital twins, new technologies, frameworks and architectures are needed. Current industrial applications are typically custom-built and tailored to the specific operational environment. With the advent of the industrial internet, new systems are built based on reference architectures, such as IIC's Industrial Internet Reference Architecture[9], allowing a certain degree of interoperability at the system-component level. Adding digital twins requires digital twin interoperability.

Interoperability of digital twins comprises two scopes:

1. *Within a digital twin system*[10]: A typical industrial application usually manages many pieces of equipment of different types, with their requisite digital twins. However, the operators and software vendors usually do not possess the necessary background to build these digital twins by themselves. On the other hand, the original equipment manufacturers (OEMs) typically have deep knowledge about the equipment they design and build, and thus are ideal contributors to provide digital twins of the equipment they build,[11] ideally along with the delivery of the physical equipment in a dual delivery scheme (see Figure 3-3). These digital twins, including their data models and computation models delivered by the OEMs, must be operable in the application systems built for the specific operational environment, ideally 'plug-and-play,' runnable with data from the equipment in the target environment. It is cost-prohibitive for OEMs to custom-build digital twins for each of their customers' environments. This, therefore, calls for a standardized digital twin framework with standard specifications for individual digital twins. With a standardized framework, standard-based digital twins can be built once for a specific type of equipment and can be deployed in standard-conforming environments with limited adjustment.
2. *Between digital twin systems*: In large or complex industrial environments, there may be multiple industrial applications, each with a self-contained digital twin-enabled system. Interoperability between the digital twins in these systems is required to enable interaction between them.

---

[9] *https://www.iiconsortium.org/iira/*

[10] A digital twin system refers to an industrial application powered by digital twins.

[11] Other parties may also provide digital twins and they would benefit from a standardized digital twin framework.

As digital twin systems proliferate, many of them are independently designed and operated by various levels of organizations for their specfic operational environments. These systems are also increasing connected with each other, within an enterprise or inter enterprises. These digital twin systems thus forming system of systems where the inter-system interoperability is even more crucial.

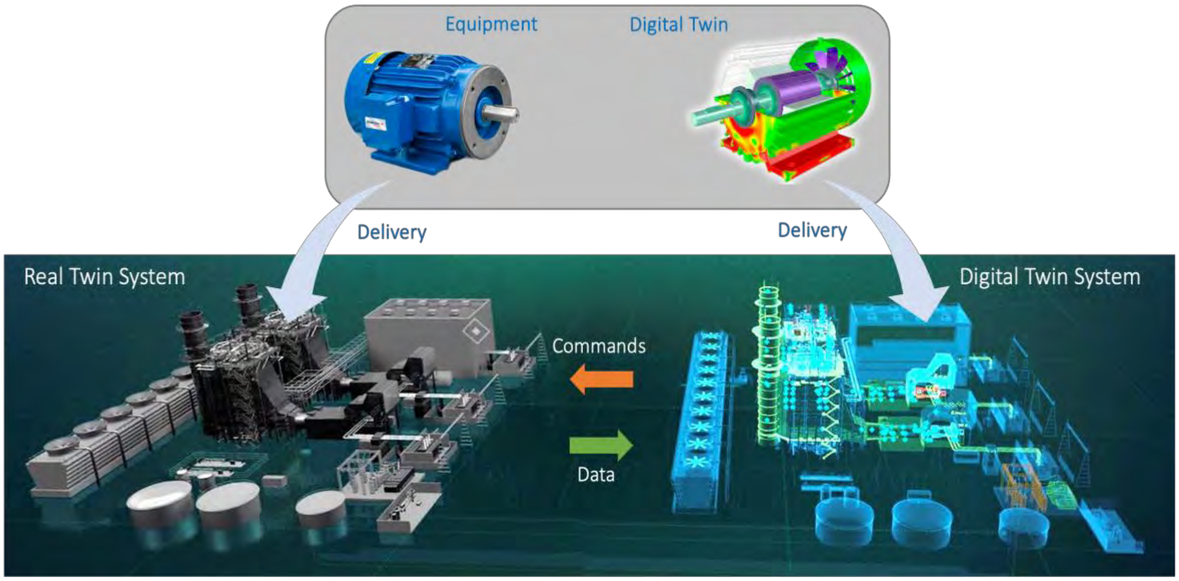


Figure 3-3: Dual delivery of equipment.

3.4 Digital Twin Standardization Approaches

Several standards-development organizations[12] have begun drafting standard architecture frameworks for digital twins. A typical approach is to define a full-stack system architecture with many functional layers or components, including equipment connectivity, data collection, preprocessing, storage and management, analytics framework and all other application-development functions. This approach is comprehensive but complex, and the resulting specifications could be challenging to implement.

An alternative approach is to focus on the core elements of digital twins and their functions to minimize the scope of the required standardization as shown in Figure 1-1. This decouples digital-

[12] These standards include, for example, the systems modeling language (SysML) *[OMG-SysML]* and Open Platform Communications United Architecture (OPC-UA) *[IEC-62541]*. A further relevant standard to specify the connector between the real and virtual world is the Asset Administration Shell (AAS) of Plattform Industrie 4.0 *[AAS-2020]* that has been submitted to the IEC 63278-1 ED1 project "Asset administration shell for industrial applications - Part 1: Administration shell structure" in IEC TC65 WG24 [IEC-63278]. The AAS is considered to be the digital twin realization of Plattform Industrie 4.0. The Industrial Digital Twin Association (IDTA, *https://industrialdigitaltwin.org/en/*) is developing sub-models of the AAS that will be submitted to the above IEC project.

twin-enabled applications into three layers, the *Support Environment Layer* at the bottom and the *Applications Layer* at the top, with a *Digital Twin Core Layer* at the middle. The core layer implements core functions representing the real world and supports applications in the top layer. The support environment layer supports the other layers by providing the requisite infrastructure.

Only the core layer is subject to standardization. The other layers interact with the core with a set of standardized interfaces. This approach gives flexibility to the implementation of the other layers. For example, the above-mentioned support functions (equipment connectivity, data collection etc.) can be readily provided by a typical IoT system. In this case, standardizing the IoT layer is not required to support interoperability of digital twins. It, therefore, allows flexibility to adopt interoperable digital twins to more IoT systems.

This results in a versatile core standard with reduced scope and complexity that supports broad industrial application systems, independent of architecture and technology.

3.5 DIGITAL TWIN DEVOPS

With the prevailing trend in application DevOps[13], it is common to see industrial applications implemented based on certain technology platforms, including some industrial platforms as a service (PaaS). These platforms typically provide connectivity to equipment and real-world environments, perform data collection, pre-processing, storage and management, enabling application DevOps including micro-service support.

[13] The combination of software development (dev) and operations (ops). For precise definition refer to *https://csrc.nist.gov/glossary/term/development_operations*.

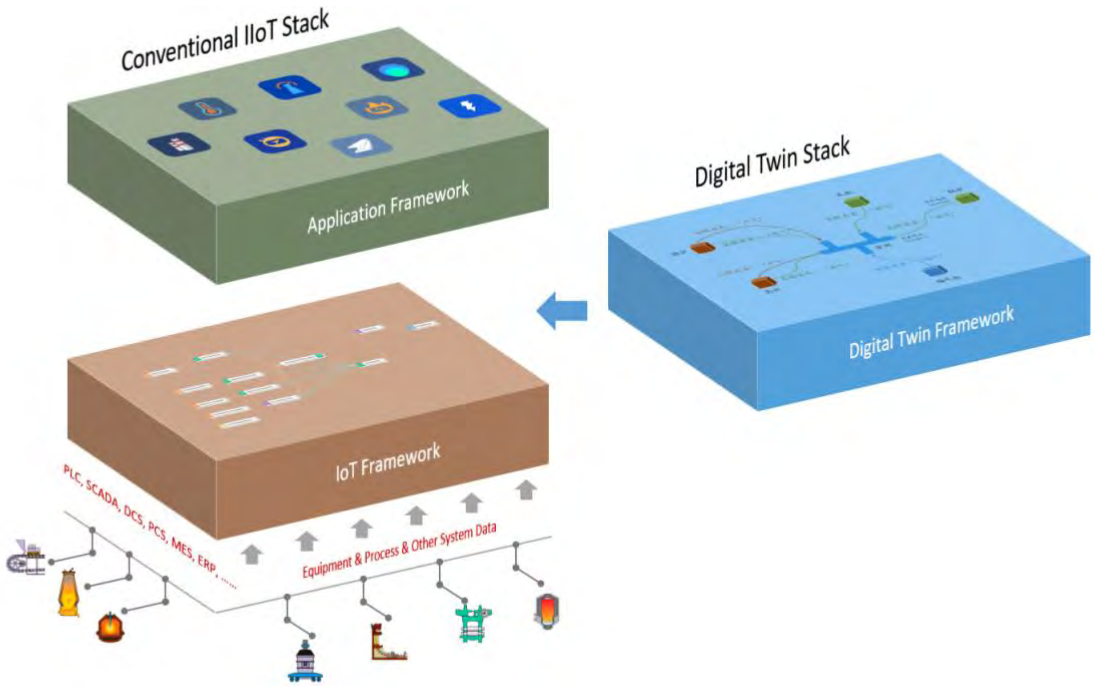


Figure 3-4: Combining digital twin with IIoT.

The purpose of these platforms is to enable and simplify application development by providing proven system architectures and ready-to-use common functionalities as platform services thus reducing the complexity of application development. For example, a common IIoT application platform may combine an IoT (connectivity, data management functionalities) framework with an application DevOps framework (left parts in Figure 3-4).

The digital twin core concept underscores interoperability and flexibility within its structure and across connected layers, enabling seamless integration of core code and modules from a wide array of sources. These sources may include commercial companies creating proprietary solutions for internal use, solution providers serving their customers, IIoT and common IT infrastructure providers such as Amazon AWS, Microsoft Azure, and Google App Engine, consultancies, open-source repositories, and even open software marketplaces like the Apple App Store or Google Play Marketplace.

Embracing this open and multi-source model within the digital twin core is particularly significant as it allows industrial experts from diverse fields to participate and contribute their expertise and specialized capabilities. To ensure smooth integration and operation, it is essential to design the core layer and its APIs with a focus on accommodating these varied component sources. This approach fosters an open, adaptable, and efficient digital twin ecosystem that promotes innovation and collaboration among different stakeholders.

3.6 DIGITAL TWIN CORE

With the approach of reducing the scope of standardization, it becomes more practical to enable interoperability for digital twins. We can define the internal constructs essential to digital twin functions and external interfaces to receive support from the underlying platform services and to provide support to the digital twin enabled applications (see Figure 3-5). Such a framework, with internal constructs and external interfaces to support essential digital twin functions is called digital twin core. It is the focus for the remaining discussion.

To summarize, a three-layer architecture model with the Digital Twin Core as its middle layer minimizes the scope of standardization for digital twins, eases adapting digital twin capabilities to a variety of architectures and maintains flexibility for selecting technologies to implement these capabilities. This three-layer architecture comprises an application layer, digital twin core and the support environment.

- The application layer is where the business logic is implemented. In solving specific operational problems, the application layer requires information and analytics about the states and behaviors of the real-world entities provided by the Digital Twin Core layer.
- The Digital Twin Core layer implements the digital representation of, or the modeling of, the real-world system or component.
- The Support Environment Layer includes technological frameworks that support the middle- and upper-layer components. From the perspective of supporting the Digital Twin Core, it may include a number of support technologies.

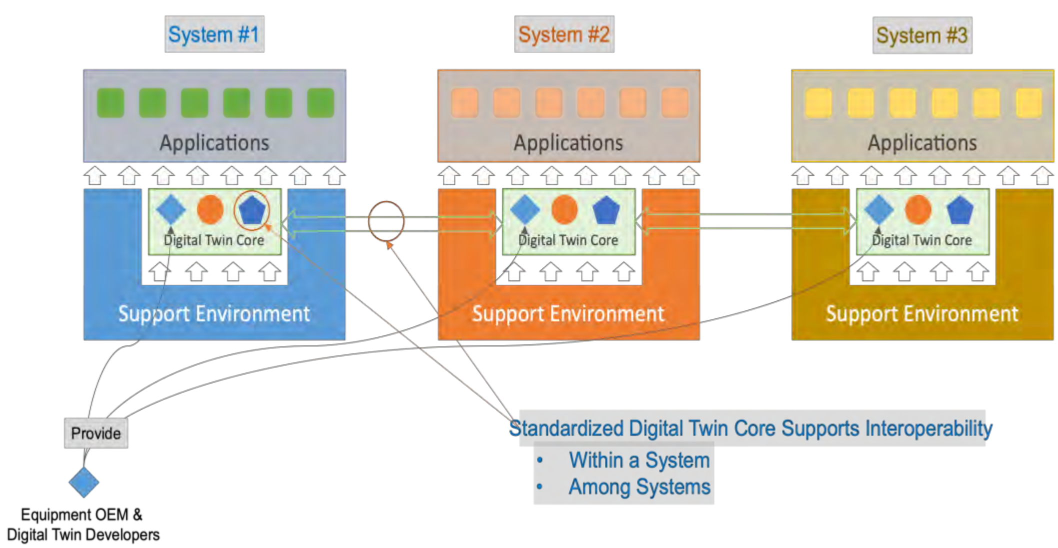


Figure 3-5: Interoperability with digital twin core.

3.7 OPEN AND INTEROPERABLE DIGITAL TWIN ECOSYSTEMS

A standardized core helps OEMs and other relevant parties deliver their respective digital twins along with the physical equipment to the operational environments, regardless of the specific architectural and technological implementation. This reduces barriers for OEMs to support their equipment with digital twins because their digital twin models can be built once and deployed in many of their customer's environments, see Figure 3-5.

Although OEMs generally have deep expertise about the equipment they make, they may not have comprehensive knowledge about how their equipment is used, especially when involving process controls in specific environments. In many cases, operators rely on external expertise for design and implementation of operational process controls. A standardized core makes it easier for domain experts, other than the OEMs or the operators, to provide their process control optimization models to be plug-and-played into the digital twin applications.

Highly interoperable digital twin systems enabled by such a core create a more open and innovative ecosystem, especially in long supply and value chains, as shown in Figure 3-6.

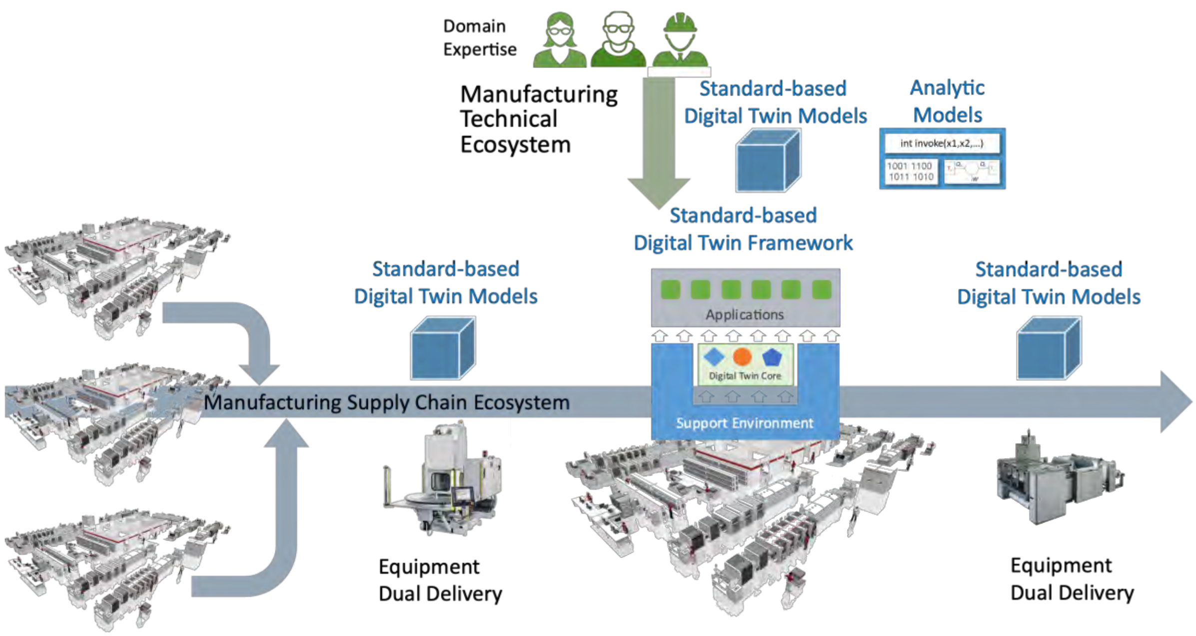


Figure 3-6: Digital twin ecosystem enabled by a standard digital twin core.

The digital twin core concept and its potentially standardized framework are not limited to manufacturing applications but can be extended to various IoT implementations across different vertical markets. For example, in the healthcare sector, an MRI machine manufacturer may possess in-depth knowledge about the device's inner workings, but they may not have the expertise to optimize its use in diagnosing patients in a clinical environment. By leveraging a standardized digital twin core, domain experts (e.g. radiologists in the MRI machine example), in additional to the OEMs or operators, can provide their process control optimization models to enhance the machine's performance in specific environments. This adaptability and flexibility of the digital twin core make it a valuable framework for a broad range of industries and applications, fostering innovation and collaboration across multiple domains.

4 Basic Requirements

To guide the development of a model for digital twin core, this section lists some common requirements[14] for digital twins in general terms. It may also serve as a reference for concrete and detailed consideration of requirements of specific digital-twin applications.

4.1 Functional Requirements

Correspondence to real-world entities: Following the digital twin definition, a digital twin shall correspond to a real-world entity (such as a physical asset or process).

- This correspondence may be one-to-one or more generally many-to-many.
- The digital twin should represent the states and their dynamics (behaviors) of the corresponding real-world entity with a defined synchronization between them, meeting the needs of given use cases.

inheritance types and instances: To maximize the reusability of digital twins, a digital twin implementation should employ concepts from object-oriented programming, by which:

- A digital-twin model-type should be defined for a type of real-world entity, such as a type of motor. For a given deployment environment, there could be multiple instances of the same type of real-world entity where the digital-twin model-type can be used as a prototype to create individual concrete digital-twin model instances, each corresponding to one real-world entity,
- A digital-twin model type should support inheritability so that a more general digital-twin model type can be inherited and extended to create a more specific digital-twin model type. This ensures consistency, encourages reuse and supports flexibility.

Digital twin templates should be provided for a given type of real-world entity as an alternative to the inheritance types and instances to simplify and standardize digital twin implementations.

Properties shall be defined for a digital twin. They can be divided into two categories:

- Description about the digital twin itself that are common to all digital twins (e.g. ID, name, creator, version, use and domain) and

[14] It is understood that this document is not a standard but contains ideas that may be taken as input to developing formal standards. Basic requirements for digital twin are part of these ideas. With this understanding and for clarity, these requirements, following convention of formal standards, are classified as mandatory or recommended.

- Descriptions about the real-world counterpart. These are domain-specific properties that may include static attributes (specification parameters for equipment) or dynamic data (reflecting the real-time states or conditions of the equipment, such as the rotational speed of a motor).

These properties:

- Shall be extensible (e.g. using inheritance) to meet the needs of given use cases and include semantic references (i.e., extra information to further clarify the meaning of a property) where needed,
- Shall contain sufficient metadata (data about the property) to ensure provenance: authenticity of the data collected from, and traceability of decisions made, and actions taken on its real-world counterpart and
- Shall be accessible through a common service interface.

However, how to store and access these properties is implementation dependent; the property values may be stored locally, or remotely accessible through references.

All *digital twin structures* shall be represented by a digital structure that includes data, models and services and can be understood by other systems. It may be restricted to properties of its real-world counterpart, or it may also incorporate a model of its dynamic behavior.

The digital structures should comprise:

Data:

- Data may be of different types: numerical (including single numbers, arrays or tensors) and text (strings or composite structural data such as JavaScript Object Notation (JSON), Extensible Markup Language (XML), and YAML Ain't Markup Language (YAML) data and data bitstreams, audio, video, image, etc.[15]
- Metadata[16] for the data shall provide sufficient semantic specification, e.g. data type, description, ownership, provenance, data unit, quality, precision, accuracy, sampling rate, value range and change rate range. Other quality attributes may be required depending on the application.

[15] This extends ISO 23247-1, Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles, section 5.3.1.3 Presentation. *https://www.iso.org/standard/75066.html*

[16] Metadata is information about specific data so that the data can be understood and used. The list of metadata types can be understood as the characteristics of the specific data, or attributes of the data. The definition of a property or attribute for a digital twin should provide the set of information about its corresponding data.

Models:

- Many different types of computation models (e.g. 3D models, simulation models, first-principal models and data-driven models) should be supported as required by the application. They may be executed by various specialized software systems.
- Properties of models may include model type, ownership, type of input/output parameters, type of software to execute it, configuration parameters and its location.
- Fidelity:[17] the model shall describe the aspects of the real-world counterpart entity relevant for the use case. The degree of fidelity may vary depending on the available computational resources and knowledge about the real-world entity.

Services:

- Each digital twin should have an API with a basic set of methods (e.g. get/set for data and other create, read, update, and delete (CRUD)-like methods).
- Generic services include access to data, data visualization and description, optional services include fault and behavior diagnosis, output prediction, prescription, and control.
- Domain and application-specific services may be defined for a digital twin.
- A digital twin may execute either locally or remotely.

Relations between digital twins:

- A digital twin may include or refer to other digital twins through unique IDs.
- All services shall account for the nesting of digital twins and their dependency on other digital twins.
- A synchronization mechanism between dependent or related digital twins may be required; this includes rules on what data is to be synchronized and when.

Connection services: These services may be provided to collect data from[18] and possibly to control the associated real-world entity depending on the use case:

- The interaction pattern of the services may include request/response or publish/subscribe.
- Data shall be transferred using standardized and secure communication protocols. It includes fault handling, data verification and validation, especially for critical commands to the real-world entity.

[17] ISO 23247-1, section 5.3.2.1.
[18] ISO 23247-1 is more specific for digital twins for manufacturing: A Digital Twin for manufacturing shall collect sensory data using sensors installed on or around manufacturing equipment.

- If needed, synchronization[19] of data between the digital twin and its real-world counterpart can be triggered by external events or real-time interactions between them on synchronized clocks. Digital twin updates may be at a lower frequency than state changes in its real-world counterpart depending on necessity or available computational and connectivity resources.

Orchestration between digital twins: Orchestration of computation models and services between digital twins, or between digital-twin components of a single digital twin, should be supported as required. This involves controlling the synchronization points (either time- or event-based) at which data is exchanged between digital twin components.

Serialization of digital twin representation: (De)serialization of digital twin representation in different formats (JSON, Resource Description Framework (RDF) or different database systems) may be required to facilitate interactions with application software or other digital twins.

Registry: Digital twin registries should be established to find digital twins with given properties or to find dependent digital twins.

4.2 Non-functional Requirements

Interoperability: A digital twin shall be easily integrated with other peer digital twins and other software systems that use or support its functionality.

Deployability: A digital twin should be deployable on a variety of platforms and infrastructures, ranging from embedded systems to data centers.

Extensibility: Digital twins should be extendable with additional properties and models.

Adaptability: Digital twins should be able to be adapted to the changes in the environment (e.g. replacement of a sensor with a new one or update of a model based on new insights).

Scalability: A digital twin system should support extensibility and adaptability at scale to maintain adequate performance when new digital twin models, more computation models or data are added, or more computation-intensive models are included. The dimensions of scalability include data volume, frequency of synchronization and the number of digital twin entities and other related components.

Confidentiality: A digital twin may incorporate or use sensitive data from one or several sources, and it may contain or generate data with restricted access. Hence, it should be possible to specify and enforce constraints on confidentiality.

[19] Data synchronization between the digital and real-world twins is a complex problem with various requirements related to data synchronization frequency, latency, consistency, accuracy, reliability, provenance and other considerations related to trustworthiness. Refer to later sections for some limited discussions on these aspects.

Trustworthiness is a set of closely related attributes of a system that ensures that its functions perform without compromise. This includes security, safety, reliability, resilience and privacy as well as verifying and validating interactions. Requirements depend on the specifics of the use case, but some level of trustworthiness is mandatory. A high-level of trustworthiness is required wherever critical real-world processes depend on the digital twin. See IIC's IoT Trustworthiness Framework Foundation[20] and IoT Security Maturity Model Digital Twin Profile[21] for further information about the considerations and concerns that must be addressed in that context.

Configurability: A digital twin should be configurable to accommodate changes in its real-world counterpart or to the available digital twin software modules. Reconfiguration should be dynamic so the digital twin can be changed during operation. The integrity and traceability of the configuration over the entire life cycle of the system should be ensured.

Customizability: A digital twin should allow for customization of user interface (UI) to make it more convenient to use and work with.

5 Digital Twin Core Element Model

Rigorous modeling is required to realize the benefits of the core, especially for maximizing the interoperability of digital twin technologies and implementations. This section outlines the basic constructs of the core using the Unified Modeling Language (UML). These models are comprehensive enough to convey the core elements without being sidetracked by non-essential details and they are a base to build more rigorous models.

Each entity (see Figure 5-1), has a minimum set of common attributes:

TypeName: The identifier of the type of an entity, of the type programmingIdentifier, such as *PulverizedCoalBoilerDigitalTwin* for digital twin of some type of pulverized coal boilers.

DisplayName: Displayable name of an entity, of the type i18nText (internationalized text), such as "*Pulverized Coal Boiler Digital Twin*" or "*Kohlenstaubkessel Digitaler Zwilling*" in German. *Identifier*[22]:

Universal identifier of an entity as a Universal Resource Identifier (URI) or Internationalized Resource Identifier (IRI) or International Registration Data Identifier (IRDI) or Digital Object Identifier (DOI). Many companies also use Universally Unique Identifier (UUID), a 128-bit label, as a unique identifier. For further details see the IIC Technical report "Identification of

[20] *https://www.iiconsortium.org/pdf/Trustworthiness_Framework_Foundations.pdf*

[21] *https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2022/06/SMM-Digital-Twin-Profile-2022-06-20.pdf*

[22] TypeName and Identifier are equivalent to class and instance names in OOAD, respectively.

Information Entities."[23] Unique identifiers are essential to be able to retrieve the correct information entities, retrieve further information about the entity (such as documentation), to link entities and define access and usage rights.

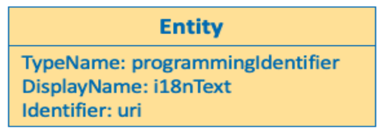


Figure 5-1: Standard entity attributes.

For a given entity, additional attributes may be defined. In the following diagrams, the standard attributes in entities are not explicitly included for the sake of brevity and clarity.

5.1 DIGITAL TWIN SYSTEM AND DIGITAL TWIN CORE

The digital twin system:

- Implements or contains digital twin functions and services that require interoperability and
- Operates independently to provide self-contained functions and services to serve certain (business) purposes.

Such a system is referred to in this document as a *digital twin system*.

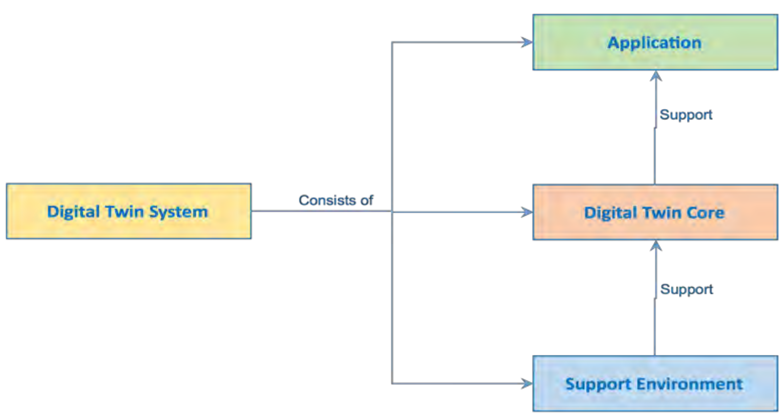


Figure 5-2: Digital twin system and digital twin core.

[23] *https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/08/Identification-of-Information-Entities-2022-08-23.pdf; https://www.isa.org/products/ansi-isa-95-00-01-2010-iec-62264-1-mod-enterprise*

In a digital twin system, the core functions of digital twin are implemented by the digital twin core entity, supported by its support environment, and in turn supports applications (Figure 5-2).

Entities

Digital twin system is an independently operated system of interest contains digital twin functions or services to support high level business application functions and services. Examples include IIoT systems and smart manufacturing systems that incorporate digital twin technologies as functions and services.

Digital twin core implements digital twin functions and services.

Digital twin core abstracts, encapsulates and provides digital twin functions and services. It can be implemented as independent software components, preferably in a form of loosely coupled services to the other components of the digital twin system or industrial application that it serves.

Digital twin core relies on computing infrastructural and IoT-related functions and services of the support environment. Examples of these functions or services include:

- Computing infrastructure and resource management systems in which the core functions and services operate and interact with the rest of the system: such as edge computing devices, servers, infrastructure as a service (IaaS), PaaS, containers and related cloud-native computing frameworks, including computing acceleration resources (e.g. Graphic Processing Units (GPUs)).
- Data storage and database systems for storing and retrieving various data required by the operation of the core functions and services, most prominently digital twin properties (see later sections) data and state data.
- Networking through which the core functional and service components interact with each other and the rest of the system.
- IoT functions include communication and connectivity to the real-world entities, data collection, pre-processing, synchronization and persistence.
- Real-world interaction or control functions including sensors and actuators in real-world entities.
- Other IT system management and security functions and services.

Support environment comprises functions and services supporting the digital twin core including those listed above and other application components.

The *application* realizes the business functions and services that fulfill the business objectives of the digital twin system. The set of business functions and services are enabled or enhanced by the core functions and services–increasingly real-world-data, analytic- and AI-driven. The applications may employ the same support environment on which the digital twin core relies.

Relations

- A digital twin system typically contains a Support Environment (a 1:1 relation), a Digital Twin Core (1:1) and one or more applications (1:1..*).
- A Support Environment supports one Digital Twin Core (1:1).
- A Digital Twin Core supports one or more Applications (1:1..*).

5.2 DIGITAL TWIN CORE INTERNAL CONSTRUCTS

The digital twin core contains internal abstract constructs and relations among them that provide the backbone in which the core functions and services manifest. A digital twin core is an aggregation of a Digital Twin Assembly, which in turn is an aggregation of Digital Twin Entities (Digital Entity in short, to correspond to Physical Entity for the Physical Twin Entity) as shown in Figure 5-3.

Entities

Digital Twin Assembly is an aggregation of digital entities. A digital twin assembly can be inherited from more abstract to more concrete definitions.

Digital (Twin) Entity represents a real-world entity that is extendable and recursive – a Digital Entity can be an aggregate of other Digital Entities.

Data Model specifies the syntactic and semantic data definition corresponding to or related to the real-world entities. It is extendable. Data access pertaining to a real-world entity will be provided by this data model. Data as the result of computation should be defined in the data model as long as it pertains to the property or state of the real-world entity.

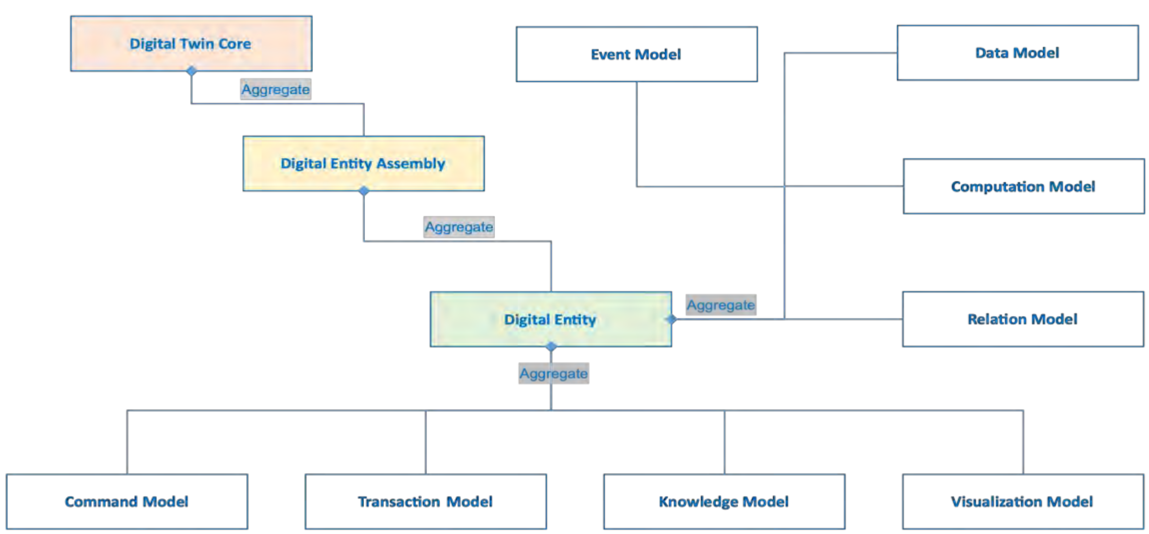


Figure 5-3: Digital twin core internal entities.

Computation Model specifies the computation definition and implementation for state and behavior (how the state changes) corresponding to the real-world entities. It specifies input and output and algorithms that generate the output. The computation model is extendable. Examples of computation models include algorithms for identification of certain patterns about the state of the real-world entity, prediction of its future states, optimization about its operations and simulation of behavior independent of current state of the real-world entity.

A computation model comprises algorithms with defined inputs and outputs. The input and output data should conform to or be mapped to the data models defined in the digital entity. Input data may come from a real-world entity (such as a sensor) or as the output of another computation model. Input and output data may be data streams.

The computation models in a digital twin are associated with its real-world counterpart under a particular operating condition and environment. The execution of a computation model can be periodic or triggered by new data (e.g. as in stream processing) or events (e.g. generated by state changes or user interaction.

Command Model: an entity that specifies the actions to be taken in the real-world entity. It may include:

- Semantics–nature of the action (e.g. open up a particular valve),
- The measure of the action (e.g. by how much),
- Metadata about the action (e.g. units of the measure) and
- Constraints of the action include valid range of measure, timing or expiration.

A digital twin may interact with its real-world counterpart to meet the safety requirements of the application. This may involve sending commands that should be verified in advance by computation models, mapped to a specific machine/device/controller/actuator in a target format and further verified by the real-world entity before execution. The corresponding edge device should check for safety and applicability in real time and may control the scheduling of command execution. Commands to a real-world entity need to comply with defined semantics. A semantic command defined in digital twin may be translated by the edge device into lower-level commands (e.g. the movement of a robot arm from position A to position B may be realized through several lower-level commands specifying trajectories and velocity ramps).

*Relation Mode*l specifies the relations by which the corresponding real-world entities related to each other. A relation may be:

- Hierarchical (in-scope-of, or be a member of, an assembly),
- Associational (customizable associative relations, e.g. supply-to, consume-from, upstream-to, downstream-to, e.g. a set of equipment connected to a pipeline, or between a machine and the product it produces) or
- Class-oriented (e.g. derived-from to indicate a specialization of a model).

The aggregation of Relation Models from all the Digital Entities assembles the real-world entities to form larger systems from sub-systems.

The aggregation of Relation Models from all the Digital Entities represents how the real-world entities are assembled to form larger systems from sub-systems (Figure 5-4)[24]. Associative relations are for example between the elements of a set of equipment connected to a pipeline, or between a machine and the product it produces.

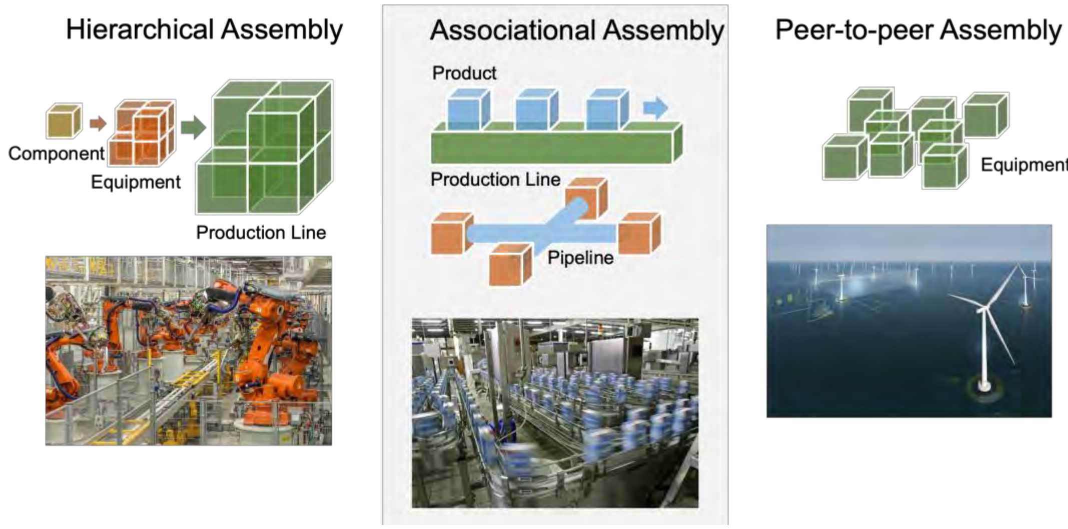


Figure 5-4: Examples of digital twin relations.

Visualization Model specifies the methods and implementations of visualization of the corresponding real-world entity, as part of the general human-machine interface. Examples of visualization models include any 2D or 3D simulated visualization, VR, AR components or metaverse. Visualization applications, in principle, may be in both the core and the applications supported by the core. The objective is to maximize the usage of re-useable, generic visualization functions in the core.

Event Model [25] specifies the events generated during the operation of the corresponding real-world entity or asset, which may include event type, time, content, suggested action, and record of action performed in response to the occurrence of each event.

[24] The Peer-to-peer assembly can be considered a special (flatten) case of hierarchical assembly.

[25] An event differs from data in that event is a piece of higher-level information about something has occurred, the generation of which could be based on lower-level data, i.e. a pressure measurement (data) has exceeded a threshold.

Transaction Model specifies the record of operational and business transactions, e.g. installation records, maintenance records, acquisition and transfer records of the corresponding real-world entity.

Knowledge Model specifies human consumable information pertaining to the real-world entity, e.g. Operational Manuals, Repair Manuals.[26]

Relations

- A Digital Twin Core aggregates zero or more Digital Twin Assembly.
- A Digital Twin Assembly aggregates one or more Digital (Twin) Entities.
- A Digital Entity aggregates zero or more Data, Computation, Command, Relation, Visualization, Event, Knowledge and Transaction Model.

In the following sections we expand the discussion on the data model described previously in Figure 5-3 and its sub-models as they are most important models in the digital twin core. Other models are left for future consideration.

5.3 DATA MODEL

The data model syntactically and semantically specifies and organizes data that represents and pertains to its real-world entity, including its properties and states (Figure 5-5).

A data model describes the semantic meaning, labels, properties, description, units, value ranges and change-rate ranges, of data such as properties and states pertaining to the corresponding real-world counterpart. The data model description is independent of the format in which the data is collected, transmitted, processed or stored with a particular technology. Data models as understood in this section are therefore positioned at Level 3 'Conceptual Layer' in the abstraction layers of an information model as described in the IIC Whitepaper 'Characteristics of IoT Information Models.'

Entities

Data Domain is the logical organization of the data pertaining to the real-world entity so the data can be easily understood, managed and accessed. It can be organized in many ways depending on the engineering or business requirements. The data domain is extensible.

Take an electric motor for example, the data can be classified into specification and operation domains. The specification domain contains data about the specification of the motor, its geometric parameters (shape), its weight, its operational limits (e.g. maximum current) etc. The operation domain contains data about its operational state, such as current, voltage, frequency,

[26] They may not be conventional "document" but searchable and interactive programs that provide the required knowledge, e.g. infographics, or even in the form of VR.

rotational speed, temperature, vibration data, all of which vary dynamically with time during operation.

Data Entity is a logical unit of data that represents one specific observation or measurement, or calculated properties of the real-world entity. It is extendable. Geographic information – observations and measurements, Open Geospatial Consortium and ISO 19156, 2011[27], specifies the generic Observation and Measurement Model (O&M Model). An observation is defined to be "an act associated with a discrete time instant or period through which a number, term or other symbol is assigned to a phenomenon." The result of an observation is an estimate of the value of a property of the real-world entity; a feature in the terminology of Geographic information – observations and measurements, Open Geospatial Consortium and ISO 19156, 2011.[28]

A measurement is an observation where the result is a numerical quantity or represented in digital form. The result may be a time series of data, or an image, or video. Take the same electric motor for example, the weight, the rotational speed of the motor, a set of vibration data over a period (the vibration amplitude along some axis changes over time and is usually analyzed together with its time-adjacent values, for example, to perform a Fourier Transformation to obtain its frequency characteristics).

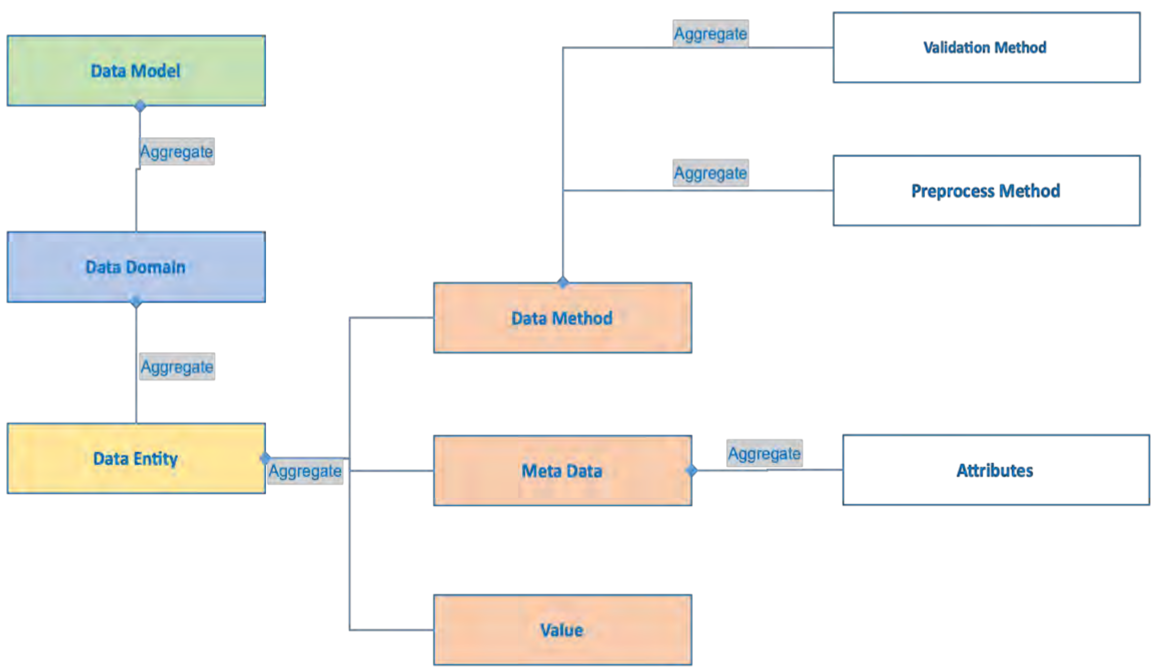


Figure 5-5: Data model.

[27] *https://portal.ogc.org/files/?artifact_id=41579*
[28] *https://portal.ogc.org/files/?artifact_id=41579*

Metadata comprises information, both syntactic and semantic, pertaining to a Data Entity so that it can be understood, processed and analyzed. Examples include the type (integer, floating point, string, array, tensor, etc.), the unit(s), internationalized human-readable name and description.

Value is the value of measurement of the Data Entity (or more generally: the result of the observation).

Attributes is a name-value pair of data structure for describing metadata, for example, units are expressed in centimeters as “Unit”: “cm.”

Data Method consists of validation and preprocessing methods to ensure the correctness and preparing for suitability of data for further analysis. It is therefore within the scope of the data entity, not part of the data analysis.

Validation Method validates the data values of data entity. Simple validations include data type validation, data value range checking, the rate (at which the data value changes over time) range checking and data anomaly detection based on a sample set (collected over time). (See 5.5 for a more detailed description).

Preprocess Method preprocesses data values of data entity including data-value smoothing, interpolation or extrapolation when data points are missing based on a sample set (collected over time), data thinning, etc. (See 5.6 for a more detailed description).

The validation or preprocess methods only concern the value of the data entity itself, not cover correlational or inter-dependence across multiple data entities. Such cross-validation should be considered at a high-level and is likely to be implemented as computation models. However, the validation or preprocess of the values of the single data unit may involve multiple data points along the time axis so the values of the single data unit can be examined correlatively in time.

Relations

- A Data Model aggregates zero or more Data Domains.
- A Data Domain aggregates zero or more Data Entities.
- A Data Entity aggregates one Meta Data, Data Methods and one Value.
- A Meta Data aggregates one or more Attributes.
- A Data Methods aggregates zero or more Validation Methods and Preprocess Methods.

5.4 Metadata Model

The Metadata Model (Figure 5-6) specifies the syntax and semantics of the value of the data entity it represents. It uses one or more attributes to provide information about the value of the data entity so it can be understood and processed. Each attribute is a name-value pair with attribute names and their corresponding values.

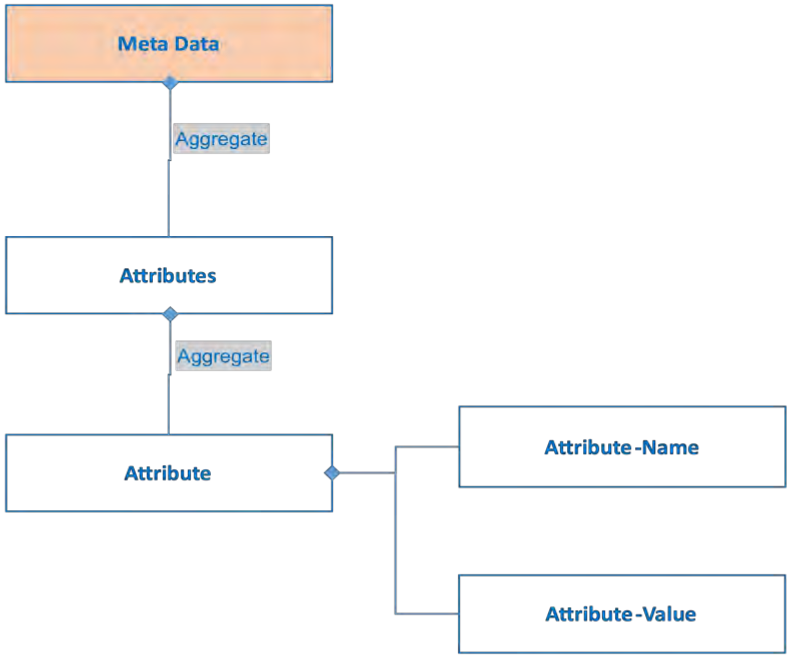


Figure 5-6: Meta data model.

Relations

- A Meta Data Model aggregates zero or one Data Attributes model.
- A Data Attributes model aggregates zero or more Data Attributes.
- A Data Attribute aggregates one Attribute-Name and one Attribute-Value.

Some common Data Attributes are listed in the table below:

Attribute-Name	Attribute-Value Type	Multiplicity	Notes
Data-Type	String, URI	1	The data type of the data entity – need to adapt to international standards
Data-Display-Name	I18n string	1	The human-readable internationalized display name
Data-Units	string	0..1	The allowed data units
Data-Unit	string	1	The currently used data unit
Data-Description	I18n string	0..1	The human-readable explanatory text about the data entity
Data-Value-Type	string	0..1	The data value type, nominal choices include real, integer, unsigned integer, string, array, tensor, JSON or other structured data, XML, binary/based-64, user-defined, etc.
Data-Label(s)	string	*	Various labels that are user-defined for classification or identification of a group of data entities
Data-Origins	array of strings	*	Records of data originality (Provenance)
User-defined-Attribute	User-defined	*	

Table 5-1: Common data types.

5.5 Validation Method Model

The Validation Method Model specifies the processing methods for validating the value of the Data Entity so that the data are correct for later analysis (see Figure 5-7).

Relations

A Validation Methods Model aggregates zero or one Validation Method model.

Common validation methods include:

- *Data Type Validator* validates that the data value is of the type specified.
- *Simple Data Range Validator* validates that the data value is within the specified range.
- *Time Correlation Validator* validates that the data value is within the expected range based on the values of data values adjacent to it in time. For instance, the rotational speed of a machine cannot change abruptly because it will violate the commonly achievable rotational acceleration given the rotational inertia of the rotating parts.
- *Data Rate Validator* validates the data rate at which the data values are collected to satisfy the specified value. (In some analyses, data should arrive at a fixed interval.)
- *Data Anomaly Detection* detects data anomaly based on physical or statistical or ML methods.[29]
- *User Defined Validator* custom validation provided by users of the system.

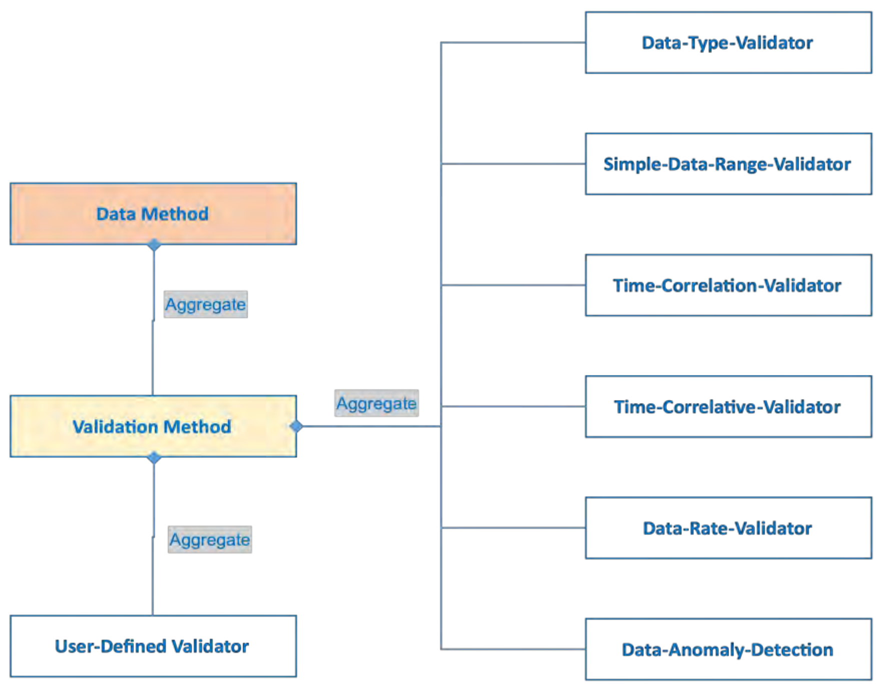


Figure 5-7: Validation methods model.

5.6 Preprocess Methods Models

The Preprocess Method Models specifies the processing methods (functions) for preprocess the value of the Data Entity so that the data are suitable for later analysis (see Figure 5-8).

[29] Though these are computation models that may involve certain algorithms, they are for preparing the data for business-oriented computation models, e.g. predicting state changes of a machine.

Relations

A Preprocess Methods Models aggregates zero or one Preprocess Method model.

Common preprocess methods include:

- *Data Unit Conversion* performs data unit conversion to the expected unit.
- *Data Smoothing* smooths data values to reduce the noise level of the data values.
- *Data-Thinning* performs data thinning processing of the data values to reduce the data density in time to achieve adequate data analysis accuracy and performance.
- *Data-Extrapolation* extrapolates data values to meet data analysis needs in case extra data points, at the end of a time range, are needed.
- *Data Interpolation* interpolates data values to provide sufficient data density or data point alignment in time or filling missing data points as required by the analysis.
- *Data Time Aggregation* statistically aggregates data values to provide aggregated values over large time ranges, for example, to provide 1-minute and 1-hour time range average values for a data measurement that is collected at a rate of 1 data point per second.
- User Defined Preprocessor: preprocessing methods provided by users of the system.

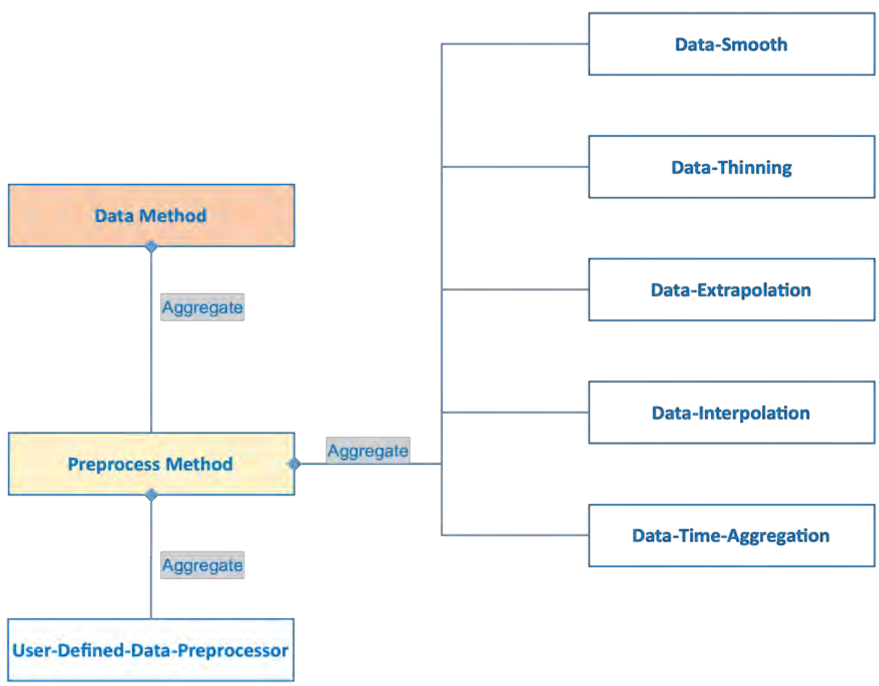


Figure 5-8: Preprocess methods model.

5.7 Digital Twin Core Services and Dependent Services

Digital twin core as a 'middle-layer' (see Figure 1-1) provides an abstraction between the real-world entities and the applications[30] and the application from the technological implementation.

Abstraction and isolation enable a high level of interoperability and fosters reusability of digital twin implementations. To achieve this, the core specifies the dependent services provided by the support environment and the services it offers to the applications. Through these services, it is possible to implement loosely coupled interfaces to connect the interdependent layers together with good flexibility and manageability of the overall system.

The following subsections outline key services that the core provides to the applications, and services from the support environment on which it relies. In an actual implementation some services can be included in the core or be provided by the dependent Support Environment.[31]

5.7.1 IoT Data Flow

The IoT data flow supports the usage scenarios in which data are collected from real-world entities, streaming or batched to update specific digital twin entities. A typical scenario is shown in the three-layer construct as depicted in Figure 5-9.

[30] Building virtual entities in computers with data attributes to represent the properties of the corresponding real-world entities and computation models to represent their dynamic behaviors.

[31] For example, the timeseries database can generally be provided by the Support Environment to the digital twin core; or be included by the digital twin core internally.

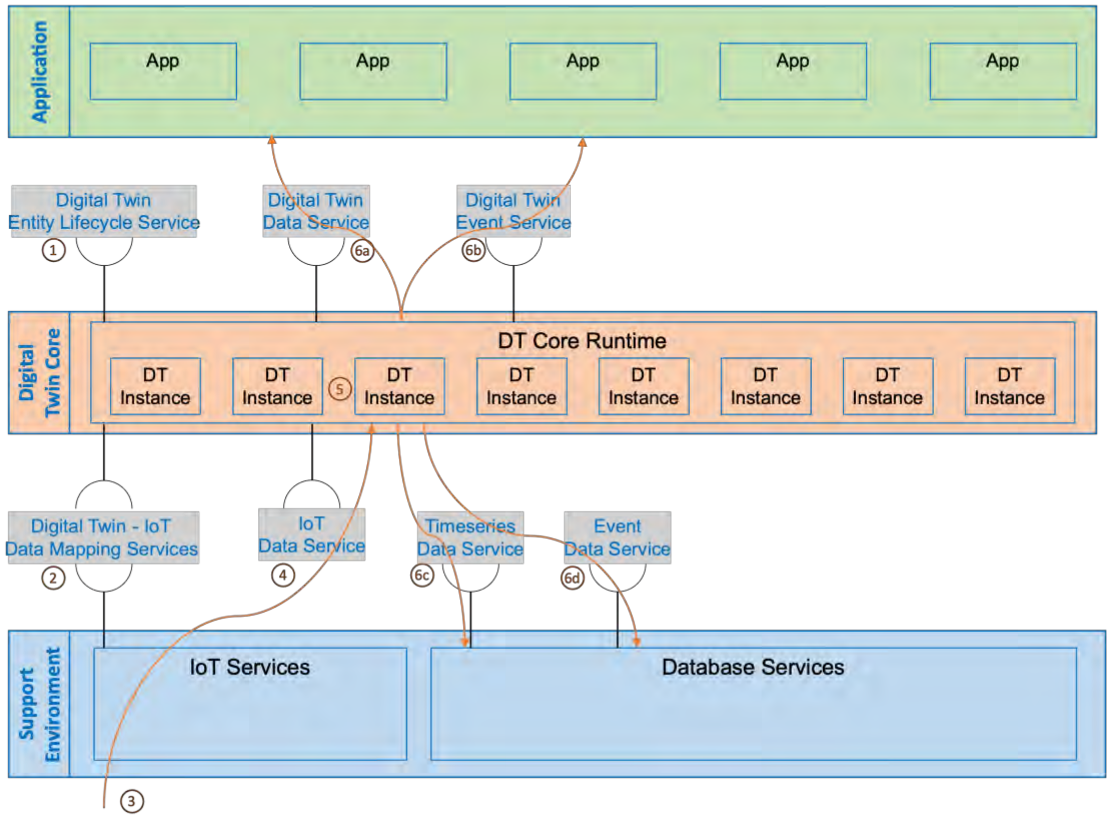


Figure 5-9: Digital twin core services and dependent services – IoT data flow.

1. A user uses an application to define digital twin model types and their instances, in correspondence to the real-world entities that are to be represented.[32] The application uses *Digital Twin Core Entity Lifecycle Service* (1) to complete the above activities.
2. A user uses an application to map the standard digital twin data definition as specified by a digital twin core implementation to the data collected from the real-world entities. Standard digital twin model types corresponding to their real-world entity types are needed to promote interoperability and reusability of digital twins. Often,[33] a data mapping service are needed to address the data adaptation needs. The mapping should cover both the semantic and syntactic aspects, including the physical, engineering or business meaning and context of the data, its data type and units. The *IoT Data Mapping Service* (2) are to be supported by both the Support Environment and Digital Twin Core.

[32] This step concerns with how to create/import a digital twin model into the Digital Twin Core. How these models are created or exported from is outside of the scope of this step.

[33] It is common at this time that the data definition in the real-world entities is not yet standardized, thus varies from one deployment environment to another. In a manufacturing environment, for example, data are typically collected from Process Logical Controllers (PLC) in which data definitions vary from environment to environment even for the same type of equipment.

3. Stateful data about the real-world (from devices for example) are collected by the *IoT Services* implemented by the Support Environment.
4. Based on the data mapping, the Support Environment forwards the data to the *IoT Data Service* offered by the Digital Twin Core, which routes the data to the respective digital twin model instances. The data can be in streaming or in batch modes.
5. The data are processed by the respective digital twin model instances and respective computation models that may be invoked in the Digital Twin Core Runtime. The trigger mechanism for the invocation could be data-,[34] event-, time- or manual-driven that can be specified by the models themselves.
6. Output of the processing and computation by the digital twin model instances, as stateful data about the real-world entities:
    a. may be pushed (e.g. pub/sub-model) to the applications that have subscribed them, through the *Digital Twin Data Service* (4) offered by the Digital Twin Core or
    b. may be stored using database capability offered by the Support Environment, through the *Timeseries Data Service* (6c) (device/equipment stateful data are of the characteristics of timeseries data, both the current value and historical values are useful and usually require persistence in computer storage to be retrieved for further analytics.[35]

As part of the processing or computation by the digital twin model instances, event or alert[36] type data about the real-world entities:

c. may be pushed (e.g. pub/sub-model) to the applications that have subscribed them, through the *Digital Twin Event Service* (6b), a capability may be offered by the Digital Twin Core or
d. may be stored using database capability offered by the Support Environment, through the *Event Data Service* (6d). (Event and alert data are like time series data that has temporal characteristics. It also requires responses, and they must be recorded).

5.7.2 APPLICATION DATA SERVICE FLOW

The application data service flow supports the usage scenarios in which the application makes data queries or updates to specific digital twin entities (see Figure 5-10).

[34] By the arrival of relevant (streaming) data.
[35] Data may be cached for performance reasons.
[36] Events are low-level records of occurrence of certain happenings that can be processed and recorded, while alerts are records containing textual information about noteworthy events that requiring human attention.

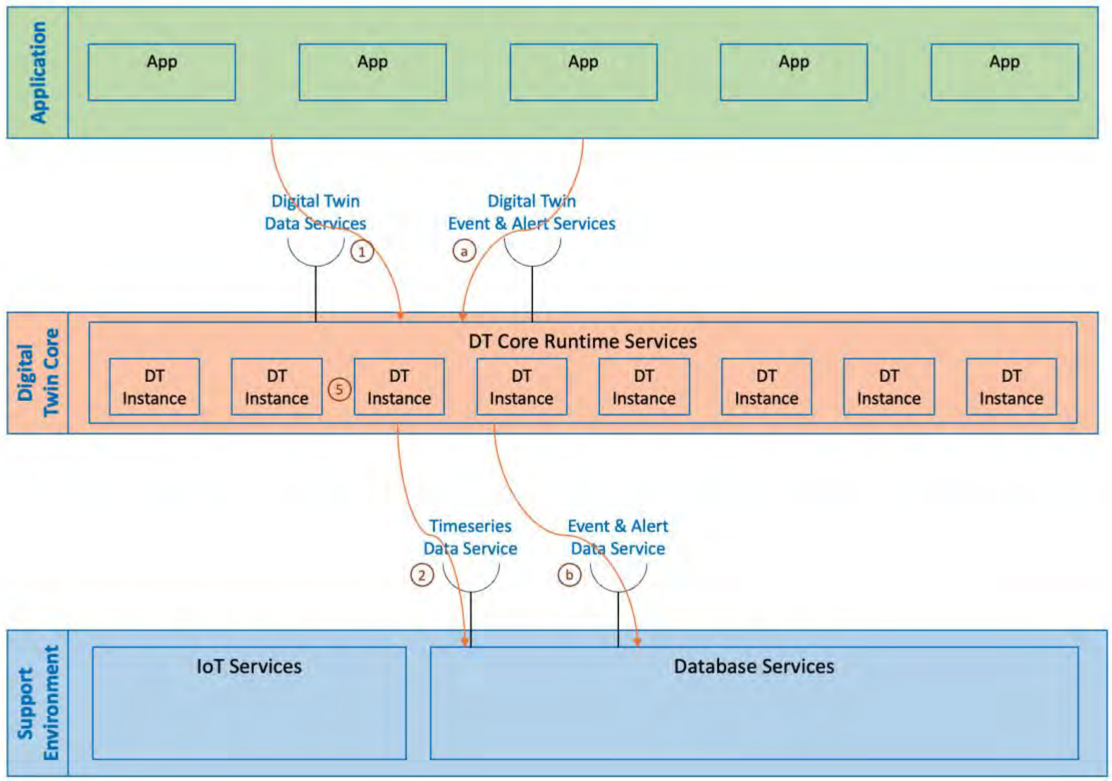


Figure 5-10: Digital twin core services and dependent services - application data service slow.

A typical scenario, using timeseries data as an example of any type of data in the core can be described as:

1. An application makes a data query or update request to the *Digital Twin Data Service* (1) offered by the Digital Twin Core, i.e. on specific data field of a specific digital twin model instance for a certain time specification.
2. The Digital Twin Core in turn makes a request to the *Timeseries Data Service* [37] (2) offered by the Support Environment; upon completion, it returns the response to the application.
    a. An application makes a data query or updates request to the *Digital Twin Event Service* (a) offered by the Digital Twin Core.

[37] Timeseries data are used as an example of this data service flow. However, other types of data contained in the Digital Twin Core concerning a specific instance of digital twin or attributes about the Digital Twin Core should be accessed and updated through corresponding standard Digital Twin Core data services.

b. The Digital Twin Core in turn makes a request to the *Event Data Service* (b) offered by the Support Environment. Upon completion, it returns the response to the application.

5.8 DIGITAL TWIN CORE DEPENDENT RESOURCES

The Digital Twin Core can be implemented as a middle layer that offers its services to support industrial applications in the application layer and relies on the services and resources from the underlying technologies and platform layer. This subsection describes computing facilities that are not normally classified as services in their narrow sense, such as computing scheduling.

5.8.1 GENERAL RUNTIME ENVIRONMENT

The Digital Twin Core can be implemented as a set of integrated software components that provide the intended functionalities. These functionalities may be implemented as modern software services,[38] interconnected but self-contained and independently managed. Distributed computing architectures are typically needed to support scalability (for performance) and redundancy (for reliability). See Figure 5-11.

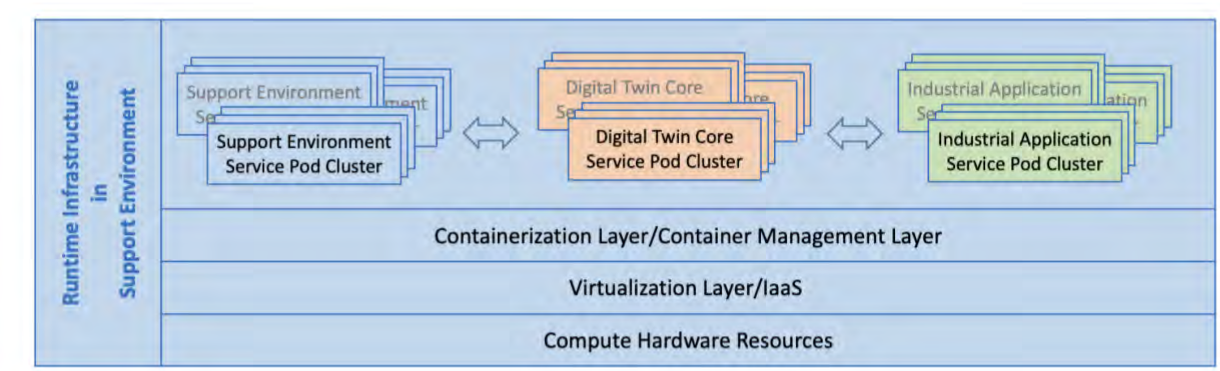


Figure 5-11: An illustrative example of Digital Twin Core in general runtime environment.

The Digital Twin Core implementation should employ modern software runtime architectures, such as containerization systems (e.g. Kubernetes), to take advantage of the computing resource manageability for distributed computing, including service scheduling, task dispatching, workload distribution, non-interruptive service upgrade, elastic scaling and agile software DevOps. The design of digital twin core middle layer with such architectural consideration eases adaptation to modern runtime environments and enhances portability from one modern runtime environment to another.

[38] Software Service is as in the sense of a service in the Service Oriented Architecture (SOA), or in its later and more specific incarnation, Micro Services.

5.8.2 ANALYTICS RUNTIME ENVIRONMENT

Analytics are essential to the usage of digital twins. Within the digital twin core, more computation models may be deployed. They will have advanced algorithms to solve ever more complex problems with higher accuracy. These computation models will demand ever increasing computing power and acceleration. For digital twins that are used to monitor and optimize productions or operations, the computation is continuous as data are streaming into the system, which adds a stronger demand on computing power.

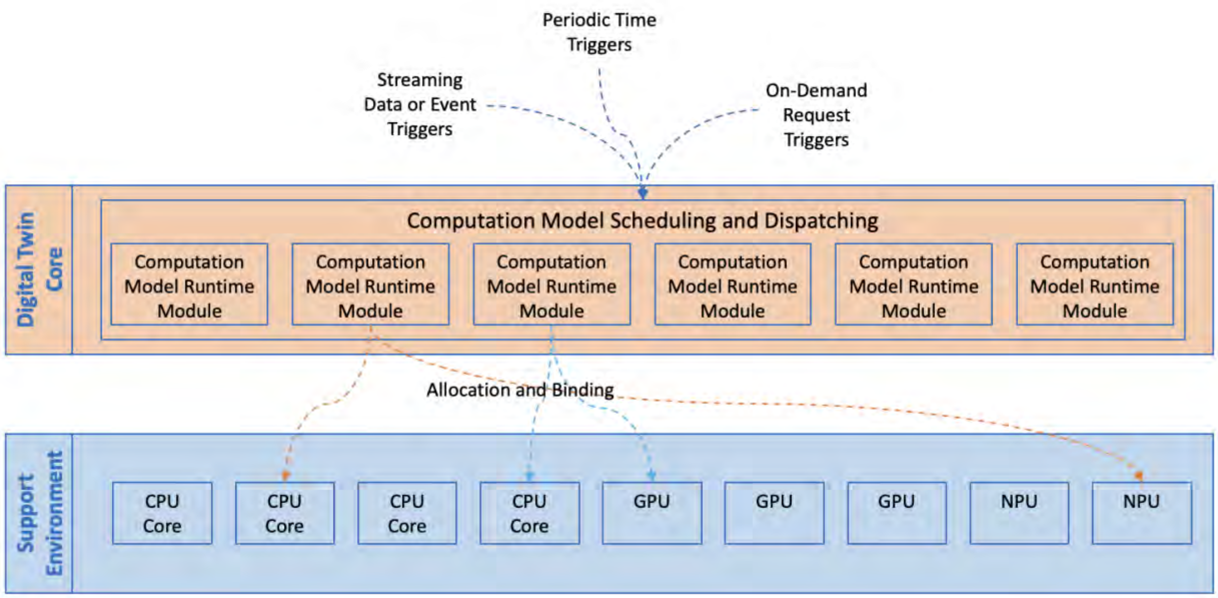


Figure 5-12: An illustration of computation model scheduling and dispatching with computing resources.

To satisfy the requirements of model computation in digital twins, the underlying runtime environment needs to provide the necessary support for scheduling, dispatching and triggering the execution of computation models, facilitating the data flows for such computations, allocating the required computation resources in terms of central processing unit (CPU) cores and memory and GPUs or neural processing units (NPUs) (see Figure 5-12).

A modern computing architecture in the Support Environment layer can support the computation needed by the Digital Twin Core layer.

5.9 INFORMATION MODEL OF CORE ELEMENTS

The IIC whitepaper 'Characteristics of IoT Information Models'[39] describes general aspects of information models that can be taken as a background guide.

[39] *https://www.iiconsortium.org/pdf/Characteristics-of-IIoT-Information-Models.pdf*

Figure 5-13 shows the layers of abstraction in an information model. The previous sections on the digital twin core are positioned largely in the Conceptual Layer, whereby reference is made to elements of Level 2, the Semantic Layer.

The Digital Twin Core is an abstract, semi-rigorous description. For an actual realization, it would need to be made formally complete and then mapped to a concrete information model covering Levels 0 ~ 2.

This can be compared with the Asset Administration Shell (AAS) of Plattform Industrie 4.0[40] where a model of a similar type is mapped to the standards such as XML, JSON, RDF, OPC UA, and AutomationML and an API is defined. This enables the introduction of proven technologies and platforms to support the Digital Twin Core (Figure 1-1), and the specification and implementation of standardized interfaces to the core elements.

Digital twin entities have in general three types of interfaces:

- Interface to the underlying platform for:
    - mapping and exchanging data streams with the real-world counterpart,
    - Issuing commands and their mappings to the real-world counterpart,
    - Using storage capability from the platform to store various type of data pertaining to the data model and
    - Using other software management functions provided by the platform.
- Interface to support applications using the digital twin core to:
    - Add/modify/delete/query/import/export/instantiation/test digital twin entities,
    - Add/modify/delete/import/export digital twin packages,
    - Query/update digital twin data and
    - Invoke digital twin computation models.
- Interface for interaction between the elements within the Digital Twin Core for:
    - Data exchange and
    - Executing computational models.

A digital twin information model may also need the following features to support a real-world solution, possibly supported by functions of the underlying platform:

- Version control,
- Identifiers for all entities and an identifier management system,
- Support for location: to determine, track, and predict the location of the real-world counterparts of digital twin entities. The location may be expressed in a real-world

[40]*https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html*

coordinate system or relative to real-world entities. This is relevant for mobile real-world counterparts. In addition, a data model may involve data with a location (e.g. for geo-spatial computational models). Cf also IIC White Paper - Characteristics of IIoT information models, Industry IoT Consortium, 2021.[41] The O&M Model [42] supports locations through metadata of the concepts Feature and Process,

- Context broker to map data models to integrate the Digital Twin Core entities into a digital twin system,
- Support for entity discovery (e.g. which core elements are available) and
- Ownership and usage rights for data and computational models: definition and enforcement. A digital twin system normally has multiple stakeholders using data from multiple stakeholders.

The digital twin core entities require a runtime environment that provides the requisite resources, supporting functions and services, including:

- Temporal management (precise clocks for real-time synchronization),
- Error handling: dealing with incorrect data for computation models, non-convergence of algorithms or excessive time or resource consumption and

Fault tolerance, redundancy, and dynamic re-configuration. There are other requirements on the runtime environment that are not unique for supporting digital twin core and are outside of scope of this report.

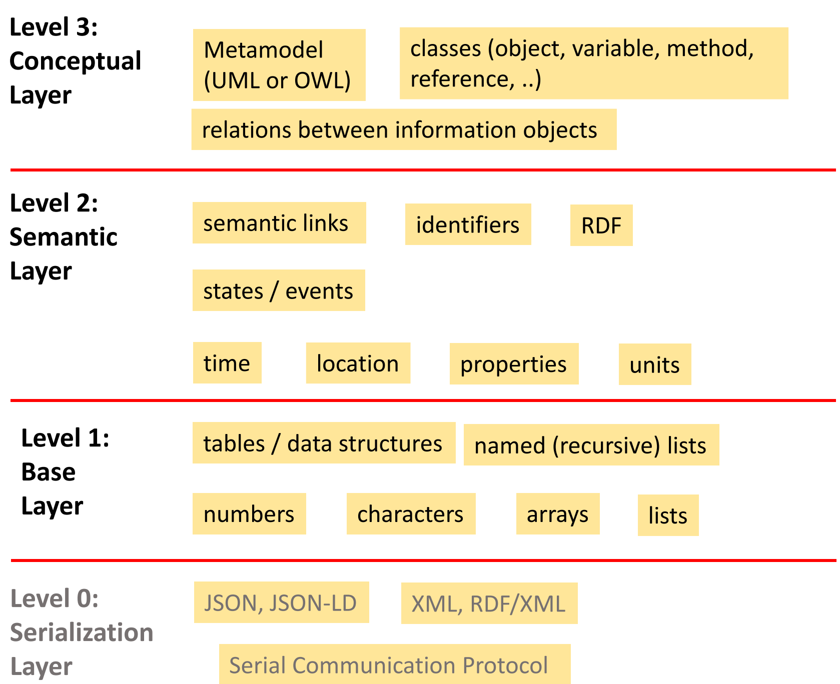


Figure 5-13: Abstraction layers in an information model [IIC-2021].

[41] *https://www.iiconsortium.org/pdf/Characteristics-of-IIoT-Information-Models.pdf*
[42] *https://portal.ogc.org/files/?artifact_id=41579*

6 INTEROPERABILITY OF CORE ELEMENTS

Because the Digital Twin Core is positioned between the Support Environment and Applications, it relies on certain features from the supporting environment to run. This feature dependency should be 'standardized', e.g. through open and extendable APIs, so that the digital twin core can be adapted to various supporting environments. Though interoperability between applications is not in the scope of this report, the vision is that digital twin core entities are designed to facilitate it.

Interoperability has aspects of scope and level.

6.1 SCOPE OF INTEROPERABILITY

The scope ranges over interoperability with real-world counterparts, within a digital twin system and between them.

Interoperability with real-world counterparts: Digital twins exchange data with their real-world counterparts, mainly for the purposes of monitoring and control. The data exchange needs to be syntactically and semantically interoperable to ensure that information coming from and sent to the real-world counterpart is properly understood and acted upon by both parties. Moreover, critical commands to the real-world counterpart may be verified and translated by intermediate equipment (such as a PLC).

Interoperability within a digital twin system: Core entities (data models, computation models, relation models) may come from one or more vendors. These entities should conform to a common standard to ensure both syntactic and semantic interoperability. Models with basic features for the same type of equipment may be supported by all vendors, but there may be vendor-specific extensions for advanced features. In these cases, interoperability also requires compatible behavior from the computation models, in functional and other characteristics, e.g. regarding its performance, timing requirements and exception handling.

There is a strong requirement to re-use existing computation models ("brownfield" modules) since they represent proven knowledge of real-world processes or advanced computational and data processing algorithms. Their integration is typically achieved by wrapping the entity in a standard API. This solves the problem of model execution and general parameterization, but again requires understanding the model semantics and behavior that then requires deeper domain knowledge.

Existing core entities may be replaced by a new entity of the same type (for example a new computation model). This also requires both interoperable interfaces and compatible (but not necessarily identical) behavior.

Intersystem interoperability between digital twin systems: Various cross-domain application systems involving digital twins may need to work together with shared data. For example, smart city, environment, transport, energy and infrastructure management need to be connected to deliver comprehensive services. Even within the manufacturing sector, there are many different types of applications used to operate and manage the manufacturing processes. These applications and their digital twins need to be integrated to deliver higher level capabilities.

Interoperability supported by the digital twin core makes it possible for different players to contribute their know-how and technologies to the implementation of the digital twins. This then allows "best of class" technologies to be deployed, reduces the reliance on single suppliers and ensures that the overall system is easier to maintain.

6.2 Level of Interoperability

The IIC Journal of Innovation - Miller, B. and Lin, S.-W.: Industrial internet: towards interoperability and composability, Industry IoT Consortium, June 2016,[43] considers three levels of general interoperability with respect to functions of a system:

- integrability: being able to integrate a function into a system. This requires that the input arguments and result types of the function are properly defined. It assumes that there is an agreed communication protocol for exchanging and storing data,
- interoperability: the functions are based on common conceptual models describing the meaning of the function, its input and result and
- composability: functions have a mutual expectation of each other's behavior in the system with respect to constraints on the input arguments and effect for the counterpart.

These levels cluster and simplify the well-known Levels of Conceptual Interoperability Model discussed in Tolk, A., Diallo, S., and Turnitsa, C.: Applying the levels of conceptual interoperability model in support of integrability, interoperability, and composability for system-of-systems engineering, Systemics, Cybernetics and Informatics, ISSN 1690-4524, Vol. 5, No [Tolk-2007].

[43]*https://www.iiconsortium.org/news/joi-articles/2016-June-Industrial-Internet-Towards-Interoperability-and-Composability.pdf*

Composability	
Level 6 Conceptual Interoperability	alignment of concepts relating to an abstraction of reality; definition of concepts with engineering tools
Level 5 Dynamic Interoperability	coping with system state changes
Interoperability	
Level 4 Pragmatic	known context and use of information exchange
Level 3 Semantic Interoperability	meaning of data exchanged
Integrability	
Level 2 Syntactic Interoperability	common and unambiguous data formats
Level 1 Technical interoperability	communication protocol for exchanging data
Level 0 No Interoperability	

Table 6-1: Levels of conceptual interoperability model.

In a digital twin system, there are typically many heterogenous components from multiple vendors and sources (such as open-source software foundries or OEMs). This complex fabric of components evolves over time in response to new requirements, the availability of improved components (such as a new simulation model) and the introduction of new IT standards for communicating and representing data. The Support Environment supporting the core components must enable the integrability level and possibly support for the interoperability level in the area of data semantics. For full interoperability of the core components, it is necessary to achieve composability so that the functional behavior of components can ensure safe, secure and resilient operation (that is, trustworthy operation) in a sustainable system of systems.

Composable components are fully specified in terms of abstract interfaces and behavior but are independent of particular implementations. They can be linked or called by higher-level components to achieve a well-defined result. Composable components need to comply with standards for the resulting component grouping to be scalable in terms of deployment, configuration, operation and management. The digital twin core approach provides a solid foundation for building composable components. Individual digital twin models and the digital twin core can be made to be composable with enhanced specification beyond syntactical and semantics of information exchange to include mutually expected behaviors.

7 Enabling Architectures and Technologies

Section 3, "Conceptual Model of Digital Twins" and Section 4 "Digital Twin Core Element Model" outlined the elements for enabling essential digital twin capabilities. These capabilities are realized through a set of services offered by the digital twin core as a number of process flows to support industrial applications. The Digital Twin Core in turn relies on certain compute resources, some of which are described in Section 5.8. The following sections provide additional details supporting the architectural, support technologies, application and implementation considerations for the Digital Twin Core.

Architectural considerations: the Digital Twin Core is a middle layer—it is not a complete and self-contained software solution; rather it is a set of technological components. This minimizes the scope over which standardization must be established, so that it can be achieved more easily. The resulting standards can then be widely adopted to support interoperability more broadly.

This middle layer approach is flexible because there are many options and variations in architecture and technology that tend to evolve rapidly.

7.1 Technologies in the Support Environment

Support environment technologies can be implemented in some form of component framework, such as simulation or ML.

Consider ML[44] or deep learning as supporting technologies. Within a specific digital twin, there may be computation models that use deep-learning algorithms to predict the behavior of its corresponding equipment. Building and running deep-learning models require a lifecycle development process supported by a set of specific tools and a runtime environment. For supervised-learning algorithms, it requires historical data for training and testing the models once created. For this purpose, support technologies may be used to collect and store the historical data and provide the specific tools and runtime environment, both open source and commercial implementations, that provides GPU or NPU hardware acceleration to train, test and run a deep-learning model. There are many open-source and commercial toolsets including GUI-based tools to specify the deep learning model, and train and test it, all supported by layers of libraries coupled with the acceleration hardware.

Typically, the environment for building and verifying models is different from the one that runs them with actual data from the real-world entities (a.k.a. inference). Both environments need to be consistent in the stack of software and hardware components that support the deep-learning

[44] *https://www.iiconsortium.org/pdf/Industrial-AI-Framework-Final-2022-02-21.pdf.*

model. The model building and running is often packaged together as a specific ML/AI framework.[45]

Many of the aspects discussed above hold true for physical or system simulation, or 3D simulation for which there are various frameworks available, both open source and commercial.

A digital twin uses technologies to achieve a certain purpose. A deep-learning framework is not an intrinsic component of the digital twin core, but a technological framework in the Support Environment. However, a deep-learning model developed for a specific digital twin for solving a specific problem is a digital twin capability and should thus be associated with that digital twin and managed by the digital twin core, within the confines of the core. The same goes for simulation or 3D models in that their development toolset and the rendering engines are considered to be part of the support environment and the individual simulation models specific to digital twins are part of the digital twin core.

Below are some important technological frameworks in the Support Environment that may be required by the digital twin core in an industrial application system.

IoT framework: To represent or model real-world "live," the digital twin core depends on connectivity and communication with the real-world entities so data can be gathered, and commands can be issued to effect changes. These capabilities of bi-directional connectivity to the real-world entities are commonly implemented in IoT subsystems.

Scale-out database framework: The digital twin core requires various databases to store and query data about real-world entities. These data comprise various types, from structured to unstructured data, from timeseries to streaming media. Some types of data could be voluminous requiring the system collecting and storing them to be easily scaled out, for example timeseries data about the state of the real-world entities with fast-moving components that may require high-frequency data collection and computation.

Modern computing runtime framework: The digital twin core would benefit from modern computing frameworks for it to be deployed, run and managed. The same framework can also support the application layer. These virtualized, distributed computing frameworks enable horizontal scaling of computing resources to support the kind of computation-demanding analytics over data gathered from the real-world entities in both volume and fidelity. The containerized environment makes it easier to deploy, run and manage the core services of the digital twin core and other services and functions for industrial applications.

Computation acceleration framework: The computation models in the digital twin core associated with specific equipment are the key elements that create values for the applications.

[45] As a concrete example, a deep-learning framework with the tensorflow's tool chains and underlined software components may consist of tensorboard (modeling building and verification GUI tool), Keras (high-level API), tensorflow (feature-rich and high-performance implementation), CUDA (GPU-acceleration toolkit) and GPUs.

The execution of these computation models will increasingly demand large computing capabilities, for example GPUs and NPUs, to accelerate advanced computation models, including analytics involving ML and AI algorithms.

ML and AI framework: Computation models in the digital twin core increasingly involve ML and AI algorithms. A framework supporting common ML and AI algorithms with toolsets for the full lifecycle of model building and execution, including data exploration, cleansing and preprocessing, model exploring, building, validating and running, are essential to the success of digital twin applications.

Physical and systems simulation framework: Simulation models are the most frequently used modeling technique in the digital twin core to perform what-if analysis, diagnostics, predictions and simulation-based optimizations. A framework that supports physical and system modeling and simulation ensures the effectiveness and trustworthiness of the modeling, synchronization and integration of digital twins. Toolsets and techniques in the framework support the whole simulation lifecycle, i.e. requirements, development, deployment, use and maintenance and retirement. Verification, validation, and uncertainty quantification (VVUQ) standards and techniques may be selected to ensure the simulation is credible and trustworthy throughout its lifecycle. It should include the interactions between simulations with the physical counterpart, with other relevant digital twin components and other existing enterprise applications.

Modern DevOps framework:[46] With the increasing demand for flexible manufacturing capabilities to support greater product variety and smaller-batch production, the manufacturing environment with corresponding industrial applications that manage and optimize the manufacturing processes will need to be more flexible to adapt and respond to changing requirements. Therefore, a flexible and agile development and operation process to accommodate enhancements of data analytics and changes of functional requirements is needed. Modern DevOps processes, methodologies, tools and frameworks streamline the collaboration between business, development and operations teams to deliver high-quality software solutions swiftly. It enables controlled tests in the production environment and a gradual rolling-out of updates to minimize the impact on the production environment.

Three-dimensional visualization, virtual reality and augmented reality frameworks: Three-dimensional (3D) visualization deals with how to render the real-world objects imitating the spatial (shape and form), material (surface texture and lighting) and movement (physical law) characteristics of these objects. These technologies have advanced a great deal in recent decades, propelled by the computer gaming industries, evolving from static single-object 3D rendering, such a part or a machine in a CAD program to more complex, dynamic and realistic

[46] Modern DevOps offers the flexibility in continuing improvement and adaptation to frequent changes of requirements, however, it is still in early stage in its application in the industrial environment in addressing strong reliability and safety requirements.

multiple objects rendering in contextual environments, giving rise to VR, AR and now the Metaverse.

These technologies aid human comprehension with intuition about the real-world, including industrial equipment. These technologies are not digital twins themselves but tools that a digital twin can leverage to enhance human comprehension of the real-world.

We envision that the support environment will provide various respective 3D rendering, VR, and AR frameworks and the digital twin core will contain various model definitions for each of the real-world entities it represents. These models can then be readily rendered by their respective frameworks on demand in the applications.

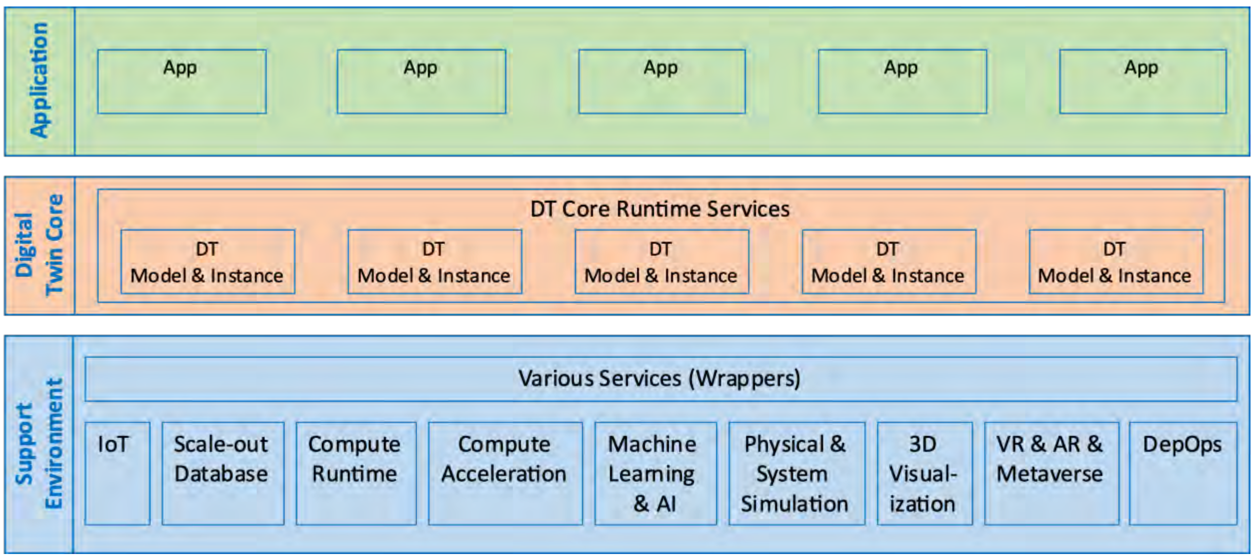


Figure 7-1: An illustration of key technologies support in a digital twin core three-tier architecture.

The capabilities from various technological frameworks (Figure 7-1) described above can be offered to, or be used, as services through some service wrappers, by the components in the upper layers. The upper and lower layers are integrated to interact with the digital twin core service APIs.

7.2 Implementation Considerations

Industrial applications range from managing, controlling and optimizing a single piece of equipment to a large collection of equipment and industrial processes. They have different levels of complexity, so there is no one-size-fits-all architectural choice.

If the application at hand requires a simple solution, the conventional monolithic application, linking in necessary software libraries may be adequate. Even for them, the digital twin core is of value in designing the software entities and components with respect to how to:

- represent the properties, states and behaviors of the equipment in the real-world,
- systematically organize various data types and models in such a representation and
- support structurally various data and workflows within the application.

It also assists in evaluating the underlying technologies that may be required in the applications, all with the right level of performance and extensibility.

If the application deals with large-scale equipment and complex industrial processes and seeks to solve complex problems, comprehensive architectural consideration with special attention to adopting proven architectural platforms is needed. Moreover, developing, deploying and operating industrial applications are not one-time activities but a continuing evolution. Industrial systems, especially those in manufacturing environments, are subject to frequent changes: manufacturing processes may be updated, materials may be changed, production lines may be altered, new environmental regulations may be imposed, etc.

The corresponding industrial applications must adapt to these changing requirements. Computation models may also require improvements to account for new operational data to increase its accuracy or precision. These models may also have to be modified to accommodate changes of manufacturing processes.

The three-layer architecture model separates the implementation concerns into encapsulated functional layers so changes can be isolated into the respective components without affecting other parts of the system. Product and process can be changed in the application layer. Computation model updates are packaged and updated only affecting the specific digital twin. Device changes are limited to updates within an IoT framework.

The digital twin core is a specific architecture pattern for conventional IIoT systems. Part of the functional architecture components in IIRA are support functions including those for collecting data from assets. The digital twin core is a more comprehensive architectural description for implementing a specific set of IIRA functions especially for data management and data analytics.

In view of the foregoing, there are several principles that are strongly recommended as part of a future-safe implementation strategy for digital twin application systems:

- Aim for an open ecosystem of technologies following an overall architectural pattern to avoid vendor lock-in, avoid dependency on proprietary platforms and retain a high level of adaptability,
- Apply standards wherever feasible, since they help reduce the effort to design components and systems, increase the likelihood of identifying compatible software and decrease the effort needed to transfer knowledge about the used technologies and train staff to maintain them. Delineate the underlying technologies and their application to

solve specific domain problems[47] and manage the semantics of all information entities since semantics are fundamental to interoperability and must keep pace with emerging application requirements and new technologies. The digital twin core and the associated dataspaces must function hand-in-hand.

There are several relevant implementation approaches (see for instance the examples summarized in "Symbiotic evolution of digital twin systems and dataspaces."[48] Based on the experience gained with these approaches, successor implementations will emerge with increasing degrees of maturity and functionality. Current IT technology makes it possible to start the journey to a digital twin core as envisaged here.

8 Typical Use Scenarios and Enabled Applications

Digital twins are changing how things work in industries from manufacturing, healthcare, energy, smart city, smart building to supply chain. A digital twin can be used to different extents ranging from simply monitoring to controlling real-world systems to enable and optimize operations. Across these levels, various analytics including descriptive, diagnostic, predictive and prescriptive analysis, can be applied to the data gathered from the real-world systems. This helps operators gain a better, more accurate and precise understanding of the conditions of the real-world entities allowing better operational decision-making. As the maturity of such decision-making based on real-world data and their computation increases, more decision-making and actions can be taken automatically reducing the need for human-in-the-loop operations.

8.1 From Monitoring to Smart Control

Digital twins may include simulation, optimization, and data analytics employing computation models based on first principle (physical or chemical laws and engineering processes), data-driven and hybrid approaches. The availability of big volume of data and advancement of relevant technologies, such as IIoT, big data and ML, enable the development of digital twins that provide insights for optimal decision making. Digital twins can support a variety of types of data analytics, e.g. descriptive, diagnostic, predictive and prescriptive analytics (Shao, G., Jain, S., and Shin, J.: Data analytics using simulation for smart manufacturing - proceedings of the Winter Simulation Conference, 2014):

[47] As a trivial example, 3D representation may be part of digital twin capabilities. A 3D model, preferably using open or standard 3D rendering specifications, for a specific piece of equipment is part of the digital twin for that equipment; a specific 3D model rendering engine is a technology implementation of a specific instance of digital twin system. It relies on the engine to render the 3D model. However, it is not necessary and best not to consider it part of the digital twin implementation but a piece of support technology. This delineation allows the flexibility to change or adapt to new underlying technologies as needed without the need of changing the domain applications (in this case rebuilding the 3D models).
[48] *https://doi.org/10.3390/automation3030020.*

- Descriptive analytics helps identify *what* happened or is happening. Digital twins with descriptive analytics provide different views of collected data such as monitoring data from device sensors to identify patterns (e.g. exceptions) and trends in such data. The output of such digital twins may be data visualization in forms of text, tables, and charts, perhaps displaying average throughput and cycle time by product types in manufacturing.
- Diagnostic analytics helps identify *why* something happened or is happening. This may include understanding the impact of the input factors and operational strategies on the performance measures. For example, the increase in cycle time of a product may be tracked down to multiple factors including machine breakdowns, worker absenteeism, material defects leading to rework, unmatched throughput in workstations in a production line and increase in priority of other products on shared machines and transporters. Sensitivity analysis may be used for diagnostic analytics.
- Predictive analytics helps identify *when* something might happen. For example, predictive analytics can estimate the performance in future periods given the current set of applicable strategies, inputs on states and conditions of the production environment. It may include estimation of cycle time and throughputs for various products based on the current strategies for order release and dispatching, scheduled material arrivals, machine and worker availabilities.
- Prescriptive analytics focuses on *how* we can make it happen and what will be the consequences. It helps identifying the strategies and inputs that will lead to desired performance. For example, prescriptive analytics may include identifying changes in input parameters and strategies that will enable cycle-time reduction and throughput increase as close as possible to the desired levels. AI/ML-driven models will play an increasingly important role.

The strategies and parameters identified by the prescriptive analytics may be used to control real-world systems. In the direct control cases where commands are issued from a system to machines without humans in the loop, stringent verification and testing must be performed before applying to the production environment to ensure operational safety. In the human-in-the-loop cases, human operators may be required to validate the actions before applying them.

Which analytics should be used or whether automatic real-time control is needed will depend on the use cases and be driven by the objective and scope of the digital twin. In addition, use cases of digital twin applications can be at any stage of the lifecycle of the real-world entity. For example, in the design stage, a digital twin can digitally represent the real-world product or system for simulation and verification; in the operation stage, a digital twin can digitally represent the real-world equipment or system for monitoring, controlling and managing the equipment or system to support operational decision makings. Data can also flow between lifecycle stages—a digital twin in the operation stage can provide feedback to a digital twin in the design stage.

8.2 TECHNOLOGY CAPABILITIES

The digital twin concept was first introduced about twenty years ago, but only recently has the advancement of intelligent manufacturing technologies, such as smart sensors, IoT, cloud computing, ML and AI, enabled digitalization and facilitated the development of digital twins in manufacturing. These technologies provide technical capabilities to support specific needs for digital twin applications in various industries. This section discusses a few typical examples of such technological capabilities.

- *3D simulation* provides realistic simulations of real-world objects. For example, in a manufacturing digital twin application, a 3D simulation can help designers and engineers visualize a computer numerical control (CNC) machining process using engineering data and real-time operational data, identify potential issues before production and imitate a CNC machine's real-world behavior by testing the cutting process virtually, i.e. determine whether the NC program is correct for producing the right product before it is downloaded into the real machine.
- *Augmented reality* can provide an interactive experience of a real-world environment where real-world objects are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory.
- *Process simulation* is used for the design, development, analysis and optimization of processes, such as chemical plants, environmental systems, power stations, complex manufacturing operations and biological processes. It could be a core component of a process digital twin.
- *Failure-mode simulation* can be used to identify possible system failures and make the system more robust before an adverse event occurs. It could be useful for predictive maintenance and equipment health, which are fundamental applications of digital twins.
- *Real-time monitoring* ensures the delivery of continuously updated information streaming at low latency. It enables the synchronization between the digital twin and its physical counterpart and ensures the near real-time status and states are modeled and analyzed for potential problems.
- *Real-time optimization* is a closed-loop application based on a steady-state model that aims at optimizing process performance in (near-) real-time. It helps systems increase performance and efficiency. Real-time optimization in a digital twin helps ensure that the optimal parameters can be derived and system performance optimized.
- *Historical data replay* is a stream of continuous snapshots that allows data to be replayed and analyzed to gain insights into the behavior, characteristics and performance of the system of interest. Digital twins with historical data replay capability can track everything in the enterprise, from processes and products, to assets, fleets and personnel. It allows review of historical data, e.g. past incidents, operational issues, maintenance data to

identify hidden issues and to find root causes for them. In addition, historical data can be analyzed to help predict the remaining life, failure or maintenance needs of assets.

- *Anomaly diagnosis* identifies rare items, events or observations that raise suspicions by differing from the expected data values. Digital twins with anomaly diagnosis capability can be used with predictive maintenance for downtime reduction. ML has been applied to detect anomalies in manufacturing processes. The trained model can detect anomalies in collected real-time data to aid predictive maintenance and downtime reduction.
- *Predictive analytics* helps *estimate* the performance in future periods given the current set of applicable strategies and inputs including historical data and executing what-if scenarios. For example, a digital twin with predictive analytics capabilities could estimate cycle time and throughputs for various products for production (re)scheduling, parts and material pre-order, maintenance scheduling etc.

8.3 BUSINESS APPLICATIONS

This section focuses on the production and operational domains[49] and digital-twin-enabled applications in terms of business applications and technical capabilities.

8.3.1 MANUFACTURING

From product design to production optimization, there exist various business applications that could benefit from implementing digital twins. For example, a real-time control application can monitor manufacturing elements using digital twins and make necessary changes to a manufacturing process near-real-time, an off-line analytics application can monitor state changes of physical systems through their digital twins and make actionable recommendations to improve the manufacturing process, and an engineering design application can use product digital twins to learn about previously manufactured products and optimize new and existing product designs:

- *Product design and engineering* applications could range from customization of product design to optimization of manufacturing systems and processes that produce the product. With digital twins, manufacturers can use and validate model-based data analytics to monitor, diagnose, predict and optimize their design and manufacturing operations.
- *Asset management* in manufacturing concerns the lifecycle of plant assets including equipment, devices, material, parts, production lines and manufacturing facilities. It involves decision making regarding asset investment, procurement, usage, maintenance and end of life. With digital twins, asset users can better manage assets' portfolios throughout their lifecycle, from cradle to grave, and get the best of all assets. It also helps asset vendors and supply-chain partners improve their products.

[49] Agriculture and mining are considered part of production and operational domains. While digital twin technologies may hold relevance in sectors like banking, finance, insurance, and education, they are not within the scope of this discussion.

- *Production planning and scheduling management:* To respond to the market demand for more customized products, manufacturing systems need to be more flexible to produce more product variants using the same resources. This includes routing flexibility and machine flexibility, i.e. an operation can be executed in more than one machine, also a machine can perform more than one operation for resource sharing.

  With a production system digital twin that represents the manufacturing environment, manufacturers can collect data from shop-floor systems such as production equipment, MES and ERP systems to monitor and analyze the current status of the production system for identifying possible fluctuations in manufacturing resources (i.e. material, labor and equipment) while satisfying customer demand and inventory level. The digital twins use the knowledge from data modeling and analysis to enable demand-driven, on-time delivery, resource optimization, cycle-time reduction, and inventory-cost reduction dynamically in a timely manner.

- *Energy management* in manufacturing concerns the tracking and optimizing energy consumption to conserve energy usage during manufacturing operations and within factory buildings. With digital twins, manufacturers can collect and analyze continuous energy consumption data, assess the energy performance, optimize energy usage by adjusting equipment schedules, set points and flows to ensure the balance and stability of energy supply and demands, improve energy efficiency and reduce energy cost. Energy management is related to climate-change emission controls. Monitoring and optimizing energy consumption in the production environment using digital twins helps reduce greenhouse gas emissions and other environmental pollution.
- *Quality management* concerns activities and tasks needed to maintain a desired level of excellence in product quality. This includes making quality policies, developing and implementing quality planning and assurance, and quality control and improvement. Using. digital twins, manufacturers can analyze relevant data in real-time to detect quality control issues as they occur, uncover the causes of the issues and determine optimal solutions to address the issues from design to production. Other stakeholders such as equipment builders and end users of the product can also benefit from the quality data of the product and the process.
- *Process management* concerns the alignment of manufacturing processes with a company's business goals, the design and application of process architectures, and proper measurement systems to evaluate the process execution. Using digital twins to represent manufacturing processes, manufacturers can manage their processes and derive continuous and sustainable process excellence. Digital twins can help manufacturers collect and analyze the real-time data throughout process execution, to identify exceptions as they occur and make optimal decisions in time by using descriptive, diagnostic, predictive and prescriptive analytics to prevent and mitigate the effect caused by process exceptions.

8.3.2 Healthcare

Digital twins help healthcare providers virtualize healthcare experience to optimize patient care, cost and performance.

- Digital twins of hospitals, doctor offices, labs and other healthcare operations can help analyze issues such as operational strategies, organization capacities and medical staffing to improve operational efficiency and optimize operational performance.
- Digital twins of patients can be created by collecting real-time patients' medical-related data. Using these digital twins, personalized care such as dynamic update of drug prescription for each patient can be achieved.

8.3.3 Energy

Digital twins could analyze the real-time status of the physical equipment and help improve its performance and efficiency. For example:

- Digital twins of wind turbines can identify signs of degradation without physical inspection of the turbine, which will save engineers' time, cost and effort. Predictive maintenance can minimize the risk of unexpected downtime and optimal performance can be achieved.
- Digital twins of the energy-generation process can provide insight into energy production, fuel consumption and CO_2 emissions. The digital twin improves process efficiency, reduces fuel usage for the same amount of energy generated and achieves plant's sustainability goal.

8.3.4 Smart City

Digital twins in a smart city allow data sharing across the city ecosystem. Data regarding terrain, communities, buildings, infrastructure, traffic and IoT devices are collected and analyzed to support decisions enabling the city to be smarter, more sustainable, and more resilient.

- Digital twins of existing cities can be used to perform long-term urban planning to help improve energy consumption, traffic flow, pollution control, safety, livability and various measures of efficiency.
- Digital twins of new city design can be used to test design ideas by simulating various design scenarios. The smartest, most sustainable, and optimal design can be selected to enable a high quality of life for citizens of the city.

8.3.5 Smart Building

The applications of digital twins in a smart building support decisions enabling the building to be smarter, more sustainable and more user friendly.

- Digital twins of a smart building can provide a complete digital model of the entire building system, including both IT and operational technology subsystems. In addition, the functionalities of the building, the policy for using the building, and the interaction of the users of the building can be modeled within the digital twins.

- Digital twins for building maintenance helps service teams switch from reactive to proactive maintenance. By analyzing relevant data and measurement of key performance indicators of the building, preventive maintenance can be performed. It can also help in improving efficiency and occupant satisfaction.

8.3.6 SUPPLY CHAIN

The applications of digital twins in the supply chain and logistics domain help enterprises better forecast supply chain dynamics, improve their resiliency and enhance their ability to respond to changes. Supply chains' behaviors can be better understood, abnormal situations can be predicted and an optimal action plan can be created.

- Digital twins of supply chains can provide logistic companies with visibility and traceability across supply-chain tiers to ensure supply chains operate at maximum effectiveness. Digital twins can monitor the supply chain to identify possible disruptions early, which results in better mitigation strategies to reduce any effects of any disruptions.
- Digital twins of transportation networks can analyze the traffic information and alternative routes. The best distribution routes and most efficient inventory locations can be determined.

9 RELEVANT STANDARDS

Digital twins are complex systems involving many aspects, e.g. data collection, information modeling, data communication, architectural framework, systems integration, simulation modeling and control. There are many existing standards covering these aspects that can be applied in the development of digital twins. Some standards focus on specific industry domains such as smart city, building and constructions, manufacturing and energy. Standards also enable the composability and reusability of digital twin components and make implementations of digital twins more effective and more economical.

Digital twin core is a high-level architectural consideration of digital twins. Its implementation can leverage many of the existing standards in one aspect or another. This section identifies a list of relevant standards specifically developed for digital twins of different domains as well as existing standards from various functional categories. Moreover, there are several digital twin specific standards and a few standards that are under development and evolving. This section is not an exhaustive list. Most of the listed standards are mainly for the manufacturing domain.

9.1 DIGITAL TWIN STANDARDS

- *ISO 23247, Digital Twin Framework for Manufacturing* provides generic guidelines, a reference architecture and a framework for case-specific, digital twin implementations. The standard supports the composability of models and interoperability among digital twin modules. It also provides examples of data collection, communication, integration, modeling and applications of relevant standards.[50]

- *IEC 62832-1: 2020, Digital Factory Framework* defines a framework to establish and maintain the digital representation of production systems throughout its life cycle. A consistent exchange of information between all processes and partners can be achieved with the support of the framework. Information then becomes understandable, reusable and exchangeable throughout the production system life cycle.[51]
- IEC TC65 WG24 is developing *IEC 63278-1 ED1 project "Asset administration shell for industrial applications - Part 1: Administration shell structure*" for the asset administration shell. The Asset Administration Shell (AAS) of Plattform Industrie 4.0 *[AAS-2020]* has been submitted to this project. Plattform Industrie 4.0 realizes digital twins using the AAS. The draft specifies the connector between the real and virtual world and includes a model of the shell covering the fundamental concepts: Asset, Sub-model and ConceptDescription. Identifiers are defined for all elements in the model, concept descriptions and property definitions of external repositories such as ECLASS or IEC CDD are referred to. Mappings of the AAS model are specified for several widely used information models in the production domain: XML, JSON, RDF, OPC UA and AutomationML.
- *IEEE P2806:2019, System Architecture of Digital Representation for Physical Objects in Factory Environments* is a standard for supporting the creation of digital factories. It describes the objective, components, data sources required and procedure of digital representation in factory environments.[52]
- *ISO/IEC 21823-1:2019, Internet of things (IoT) - Interoperability for IoT systems - Part 1: Framework* provides an overview of interoperability and a common understanding of interoperability of IoT systems and the various entities within them. It enables IoT systems to be built so that the entities of the IoT system can exchange information and use the information efficiently. It also supports peer-to-peer interoperability between IoT systems.[53]
- *ISO/IEC 21823-2:2020, Internet of things (IoT) - Interoperability for IoT systems - Part 2: Transport interoperability* specifies a framework and requirements for transport interoperability to enable the construction of IoT systems with information exchange, peer-to-peer connectivity and seamless communication between different IoT systems and among entities within an IoT system.[54]
- *ISO/IEC 21823-3:2021, Internet of Things (IoT) - Interoperability for IoT systems - Part 3: Semantic interoperability* provides:

[50] *https://www.iso.org/standard/75066.html*
[51] *https://webstore.iec.ch/publication/65858*
[52] *https://standards.ieee.org/project/2806.html*
[53] *https://www.iso.org/standard/71885.html*
[54] *https://www.iso.org/standard/80986.html*

  - Basic concepts for IoT systems including requirements of the core ontologies for semantic interoperability,
  - Best practices and guidance on how to use ontologies to develop domain-specific applications, including allowing for extensibility and connection to external ontologies, cross-domain specification and formalization of ontologies to provide harmonized utilization of existing ontologies,
  - Relevant IoT ontologies along with comparative study of the characteristics and approaches in terms of modularity, extensibility, reusability, scalability, interoperability with upper ontologies and
  - Use cases and service scenarios that exhibit necessities and requirements of semantic interoperability.[55]
- *ISO/IEC 21823-4:2022, Internet of things (IoT) - Interoperability for IoT systems - Part 4: Syntactic interoperability* describes five facets for IoT interoperability: transport, semantic, syntactic, behavioral and policy. It includes specifications from a syntactic viewpoint: how to achieve syntactic interoperability among IoT devices and a framework for processes on developing information exchange rules related to IoT devices.[56]
- *ISO/IEC 30161:2020, Internet of Things (IoT) - Requirements of IoT data exchange platform for various IoT services* specifies requirements for an IoT data exchange platform for the:
  - Middleware components of communication networks allowing the co-existence of IoT services with legacy services,
  - End-points performance across the communication networks among IoT and legacy services,
  - IoT-specific functions allowing the efficient deployment of IoT services,
  - IoT service communication networks' framework and infrastructure and
  - IoT service implementation guideline for the IoT data exchange platform.[57]
- *ISO/IEC 30141:2018, Internet of Things (IoT)* - Reference Architecture is an IoT reference architecture using a common vocabulary, reusable designs and industry best practices. It starts with collecting the most important characteristics of IoT, abstracting those into a generic IoT conceptual model, deriving a high-level systematic reference with subsequent dissection of that model into five architecture views.[58]
- *ISO/IEC/IEEE 15288:2015, Systems and software engineering* - System life cycle processes establish a common framework of process descriptions for describing the life cycle of systems. It defines a set of processes and associated terminology that can be applied at any level in a system's structure. They can be applied to manage the stages of a system's

[55] *https://www.iso.org/standard/83752.html*
[56] *https://www.iso.org/standard/84773.html*
[57] *https://www.iso.org/standard/53281.html*
[58] *https://www.iso.org/standard/65695.html*

life cycle. It also provides processes that support the definition, control and improvement of the system life cycle processes. [59]

- *ISO/IEC 30147:2021, Internet of Things (IoT)* - Methodology for trustworthiness of IoT system/service provides system life cycle processes to implement and maintain trustworthiness, supplementing ISO/IEC/IEEE 15288. They are applicable to a wide range of application areas.[60]
- *Microsoft Digital Twin Definition Language (DTDL)* - A language for describing models and interfaces for IoT digital twins. DTDL is based on JSON-LD and is programming-language independent. DTDL is used in different Microsoft services such as IoT Hub, IoT Central, and Azure digital twins, it is also used to represent device data in other IoT services such as IoT Plug and Play. DTDL covers the resource description and not resource discovery and access. Resources (interfaces) contain telemetry, properties, commands, relationship, and components.[61]

9.2 Data Collection and Device Control

The American National Standards Institute (ANSI) *MTConnect* supports interoperability by providing a vocabulary for manufacturing equipment, making possible structured contextualized data and avoiding proprietary formats. Data sources of MTConnect in production include equipment, sensor packages and other factory-floor hardware.[62]

- *OPC-UA* is a platform-independent standard used to communicate by sending messages between clients and servers over diverse networks with syntactic interoperability.[63]
- *MTConnect-OPC UA Companion Specification* ensures interoperability and consistency between MTConnect specifications and the OPC Unified Architecture (UA) specifications, and devices and software that implement those standards. The specifications are developed by the MTConnect Institute and OPC Foundation via a joint working group.[64]
- *ISO/IEC 20922, Information Technology - Message Queuing Telemetry Transport (MQTT) v3.1.1* is a data protocol to support messaging transport of client and server for publishing and subscribing. It is open and simple in its design so it can be implemented easily. It is ideal for use in machine-to-machine (M2M) communication and in IoT contexts.[65]

---

[59] *https://www.iso.org/standard/63711.html*
[60] *https://www.iso.org/standard/53267.html*
[61]*https://learn.microsoft.com/en-us/azure/digital-twins/concepts-models#digital-twin-definition-language-dtdl-for-models*
[62] *https://www.mtconnect.org/*
[63]*https://opcfoundation.org/developer-tools/specifications-unified-architecture#:~:text=OPC%20Unified%20Architecture%20Specification,more%20secure%20and%20scalable%20solution*
[64] *https://www.mtconnect.org/opc-ua-companion-specification*
[65] *https://www.iso.org/standard/69466.html*.

- *ISO/IEC 17826, Information Technology - Cloud Data Management Interface (CDMI)* focuses on cloud storage and management, specifying how to access stored data and how to manage it. [66]
- *ISO/IEC 27000-series, Information Security Management* focuses on information security management. The standard provides best practice recommendations for an overall Information Security Management System (ISMS).[67]
- *ISO/IEC 27033-series, Information Technology-Security Techniques -Network Security* guide on security aspects related to information system networks. The standards include management, operation, use and interconnections of information system networks.[68]
- IIC *Industrial Internet Security Framework* provides a common security framework and an approach to assess cybersecurity in Industrial Internet of Things systems, including data collections and edge devices.[69]
- *ISO 13374 series, Condition Monitoring and Diagnostics of Machines - Data Processing, Communication and Presentation* supports condition monitoring and diagnostic of machines by providing the basic requirements for open software specification.[70]
- IEC *62453, Field Device Tool (FDT) Interface Specification* defines the concept and relevant terms important for FDT. FDT is used for standardizing the interfaces for communication and configuration of field equipment and host systems.[71]
- *ISO/IEC/IEEE 21450, Information Technology - Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats* defines the functions to be performed by a transducer interface module (TIM) and the common characteristics for all devices that implement the TIM.[72]
- *ISO 14649 -201; Industrial Automation Systems and Integration – Physical Device Control – Data Model for CNC Controllers - Part 201: Machine Tool Data for Cutting Processes* is a standard for technology-specific machine tool description. It defines data elements needed to represent manufacturing and machine characteristics.[73]

9.3 DATA AND INFORMATION MODELING

- The *American Society of Mechanical Engineers (ASME) Y14.26M, Digital Representation for Communication of Product Definition Data* focuses on the representation and

[66] *https://www.iso.org/standard/70226.html*
[67] *https://www.iso.org/standard/73906.html*
[68] *https://www.iso27001security.com/html/27033.html*
[69] *https://www.iiconsortium.org/iisf/*
[70] *https://www.iso.org/standard/36645.html.*
[71] *https://www.fdtgroup.org/tag/iec-62453*
[72] *https://doi.org/10.3390/automation3030020*
[73] *https://www.iso.org/standard/60042.html*

communication of data to define products. It supports the exchange of product data developed in computer-aided manufacturing systems.[74]

- *ISO 10303, Automation Systems and Integration - Product Data Representation and Exchange*, also known as Standard for the Exchange of Product model data (STEP), supports the exchange of product manufacturing information.[75]
- *ISO 10303-IR 105, Automation Systems and Integration - Product Data Representation and Exchange - Part 105: Integrated Application Resource: Kinematics* focuses on the representation of kinematics information of a mechanical product.[76]
- *Quality Information Framework* (QIF) is a standard for creating digital threads. It applies to product design, manufacturing and quality inspection. It relies on the XML standard and contains a library of XML schemas. It supports data integrity and interoperability in implementing model-based enterprise and IoT.[77]
- *ASME B5.59-2, Information Technology for Machine Tools Part 2* defines the properties needed to describe machine tools used for milling and turning.[78]
- *ISO 13399, Cutting Tool Data Representation and Exchange* contains a model and a reference dictionary to represent cutting tools. EXPRESS schema is used for the product description and product files can be generated according to the schema.[79]
- *ISO/IEC 20005:2013, Information Technology-Sensor Network Services and Interfaces Supporting Collaborative Information Processing in Intelligent Sensor Network* is used for collaborative information processing (CIP). It defines services and interfaces supporting CIP in intelligent sensor networks.[80]
- *ISO 16400, Equipment Behaviors Catalog (EBC)* defines a template and rules for describing behaviors of equipment, such as state transition and time series of operation results, that are produced because of machine activities to be registered in the common repository. It specifies the methodology to construct catalogs of equipment behavior to be used to plan and analyze production system performance.[81]
- *IEC 61360-series, Standard Data Element Types with an Associated Classification Scheme for Electric Items* provides a basis for the unambiguous definition of data element types of all elements of electrotechnical systems. It ranges from basic components to sub-assemblies and full systems.[82]

---

[74] *https://standards.globalspec.com/std/437642/ASME%20Y14.26M*
[75] *https://www.iso.org/standard/72237.html*
[76] *https://www.iso.org/standard/64294.html*
[77] *http://qifstandards.org/*
[78] *https://www.asmestandardsonline.com/asme-b5-50-2009-pdf-download.html*
[79] *https://www.iso.org/standard/36757.html*
[80] *https://www.iso.org/standard/50952.html#:~:text=ISO%2FIEC%2020005%3A2013%20specifies,common%20service%20interfaces%20to%20CIP*
[81] *https://www.iso.org/standard/73384.html*
[82] *https://webstore.iec.ch/publication/28560*

- *IEC 61804-series, Function Blocks (FB) for Process Control and Electronic Device Description Language* (EDDL) can be used to describe the characteristics of devices. EDDL creates EDD files.[83]
- *AutomationML* (Automation Markup Language) is a neutral data format based on XML that can store and exchange information for plant engineering, connecting heterogeneous modern engineering tools. Disciplines include mechanical plant engineering, electrical design, human-machine interface development, PLC and robot control.[84]
- *IEC 62714 series, Engineering Data Exchange Format for Use in Industrial Automation Systems Engineering - AutomationML* supports exchange of engineering data in industrial automation systems. The format defined is based on AutomationML. This standard integrates engineering tools, and it is applicable for various disciplines, such as process engineering, process control engineering and PLC programming.[85]
- The *Core Manufacturing Simulation Data* (CMSD) standard supports the exchange of data between simulation and other manufacturing applications.[86]
- *ISO 15531, Industrial Automation Systems and Integration - Industrial Manufacturing Management Data* supports information exchange between software applications in production activities including planning, scheduling, simulation, control and execution.[87]
- *ISO 13584-42:2010, Industrial automation systems and integration - Parts library - Part 42: Description methodology: Methodology for structuring parts families* specifies how to characterize parts and their properties independently of any supplier-defined identification.[88]

9.4 Digital Twin Modeling

- The *High-Level Architecture* (HLA) supports data communication and time synchronization of distributed simulation systems.[89]
- *Predictive Model Markup Language* (PMML) is used to develop predictive and descriptive models and to represent pre- and post-processed data. PMML is based on XML and supports the representation of data-mining models. Examples include neural networks, decision trees, Gaussian progress and Bayesian networks.[90]

---

[83] *https://webstore.iec.ch/publication/60628*.
[84] *https://www.automationml.org/wp-content/uploads/2021/06/AutomationML-Brochure.pdf.*
[85] *https://webstore.iec.ch/publication/32339*
[86]*https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=31457&PortalId=0&TabId=105*
[87] *https://www.iso.org/standard/71064.html*
[88] *https://www.iso.org/standard/43423.html*
[89] *https://ieeexplore.ieee.org/document/5553440*
[90] *http://dmg.org/pmml/v4-3/GeneralStructure.html*

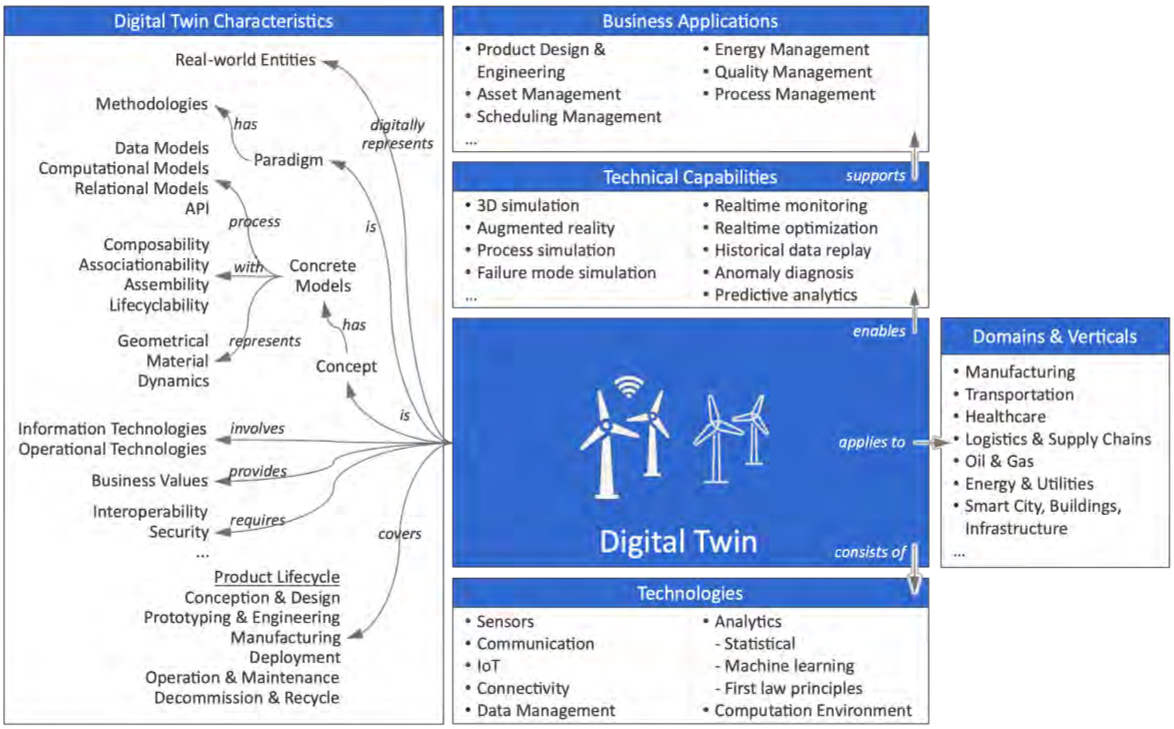
- *ASME B5.54, Methods for Performance Evaluation of CNC Machining Centers* establishes a methodology for specifying and testing the performance of CNC machining centers. It defines how acceptance-testing (runoff) should be carried out and can be used in operations to evaluate on-going capability.[91]
- *ISO 13041, Test Conditions for Numerically Controlled Turning Machines and Turning Centers* specifies, with references to ISO 230-1, the geometric tests and the applicable tolerances on CNC turning machines and turning centers with horizontal work spindle(s).[92]

With these relevant standards from various functional categories, users can select those applicable for their digital twin implementations. For example, to build a digital twin of a machine tool, ISO 23247[93] can be used as a guideline to specify the requirements, a reference architecture, and the components of the digital twin. STEP can be used to represent the design of the product. MTConnect can be used to collect the real-time operational status data of the machine tool when producing the product. OPC-UA can also be used to communicate between the machine and its digital twin and QIF can be used to represent the product quality data.

91 *http://www.leice.com/upfile/201910/2019102565834009.pdf*
92 *https://www.iso.org/standard/73489.html*
93 *https://www.iso.org/standard/75066.html*

APPENDIX A THE CONTEXT OF DIGITAL TWIN

Digital twin involves many complex and broad aspects. To understand these aspects, the context of digital twin is summarized in the figure below, a digital twin technical and ecosystem knowledge-graph. The layers in the center column relate to the architecture models in the figure below are some brief elaborations of the context of digital twin, starting with the central column.



Appendix Figure 1: Digital twin technical and ecosystem knowledge graph.

Technologies: Digital twin is supported by a set of technologies, such as sensors, communication, IoT, connectivity, data management, computation environment and analytics that in turn comprise statistical, ML and first-principal laws.

Technical capabilities: Digital twin enables a set of technical capabilities, such as 3D simulation, augmented reality, process simulation, failure mode simulation, real-time monitoring, real-time optimization, historical data replay, anomaly diagnosis and predictive analytics.

Business applications: The technical capabilities support a set of business applications that comprise product design and engineering, asset management, scheduling and planning, energy efficiency management, quality management, process control, etc.

Domains and verticals: Digital twin can be applied to manufacturing, transportation, healthcare, logistics and supply chains, oil and gas, energy and utilities, smart cities, buildings, infrastructure.

Digital twin characteristics:

- is a paradigm for digitally representing the real-world with a set of methodologies,
- can be represented by a set of concrete models comprising data models, computational models, command models, relation models, transaction models, knowledge models, visualization models, present APIs for accessing the model capabilities, present geometrical, material and dynamic property and behavior of the corresponding real-world entries to support composability, associability, assembly and lifecycle,
- involves IT and OT,
- provides business values,
- requires interoperability, security, and other characteristics (*lities) and
- digitally represents real-world entities covering full product lifecycle from conception and design, prototyping and engineering, manufacturing, deployment, operation and maintenance, decommission and recycle.

APPENDIX B DIGITAL TWIN AS A PARADIGM AND ENABLING TECHNOLOGY

A digital twin is more than just software; it is a paradigm and a set of technologies for building a digital representation of entities in the real world so we can understand and operate them better.

Digital twin as a paradigm: A paradigm sets a framework for theories and concept development. Digital twin is a paradigm for building software to manage and operate real-world entities, including large and complex industrial systems. It sets forth the approach of building software components and models that mimic and synchronize with real-world entities, and:

- *Representation*: properties and states of real-world entities can be represented as data and behaviors of real-world entities can be modeled with computation models,
- *Unity and correspondence:* properties and states (data representation) and behaviors (computation representation) for a given real-world entity can be organized as a self-contained software module–digital twin–to represent the real-world entity,
- *Synchronization*: the data representing the properties and states of the real-world entity are to be measured in the real-world and synchronized with the digital twin so they may be computed with the computation models of the digital twin and
- *Relation*: different digital twins form different relationships. Important relationships include compositional and associational relationships. Large and complex digital twins can be built with smaller and simpler digital twins.

Digital twins can build on a fusion of multiple well-known and established paradigms applied in distributed IT systems including Object Oriented Analysis and Design (OOAD) [94] , Service Oriented Architecture (SOA) [95],[96],[97] and Concurrent Automated Architecture (CAA). [98]

“Digital twin” can be seen as an extension of OOAD, benefiting from its modeling of characteristics and behavior of entities of interest to enable re-use and extensibility of the system design and implementation in representing real-world entities.

Each real-world entity type (e.g. a type of electric motor) may have multitudes of entities (e.g. many actual electric motors of the same type) that possess a common set of properties, states and behaviors shared by all entities of the same type. The digital representation of each type of real-world entities can be abstracted as a class in OOAD and each concrete instance of the entities can be represented by objects that are instantiated from the class. The properties, states and behaviors encapsulated in a class are present in all the instantiations of the class, i.e. the

[94] *https://books.google.com/books/about/Object_oriented_Analysis_and_Design_With.html?id=_8_LoQEACAAJ*
[95] *https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf*
[96] *https://www.opengroup.org/soa/source-book/stds/p2.htm*
[97] *https://www.beuth.de/de/technische-regel/din-spec-16593-1/287632675*
[98] *https://doi.org/10.1007/BFb0017625*

representation of the real-world entities of same type. Inheritance of the class allows for extensibility, such as from a general type of motor to more specific types of motors, with properties, states or behaviors modified and added with each inheritance (specialization). Associations capture the interrelations between real-world objects, such as a motor object "driving" a wheel object.

A digital twin can benefit from SOA mainly from encapsulation of implementation details and separation of concerns, and the loosely-coupling of interacting entities. With SOA, complex systems can be built from modular sub-systems or components that interact with each other through defined interfaces. The interoperability and portability of subsystems is achieved by standardizing their interfaces. When implementing digital twin systems that typically are complex systems, interacting digital twins may not be required to be running within the same software program, or the same computer, or even within the same network environment. They can be run both locally or remotely, but always capable of interacting through a common and loosely coupled application programming interface (API). One digital twin interacts with another without knowing its peer's implementation details or the details of its running environment.

The SOA approach also allows the decomposition of the system into various functional layers, each implementing defined set of functionalities independently but interacting through interfaces to enable holistic end-to-end functionalities of the system. This is exactly the approach digital twin care takes.

Because digital twins are digital representations of their counterparts in the real-world, some level of connectivity and data synchronization is required between them and the corresponding real-world entities, some in both directions. The degree of synchronization depends on the context in which the specific digital twin is used, ranging from fine-grain updates at microseconds or milliseconds to coarse scales of seconds, minutes and days. Some updates may be continuous, and others may be batched periodically, on demand or based on availability. In extreme cases, synchronization may occur once reflecting a snapshot of the states of the real-world-counterpart.[99] Here the CAA paradigm provides useful guidance on how to specify and realize such time sensitive synchronization. The CAA paradigm describes synchronization, scheduling, and time related aspects in a (distributed) IT system. For example, closed loop optimization, regulatory control and real-time monitoring require a synchronized, time-centric control of tasks in the IT system.

As a paradigm to represent real-world entities digitally, digital twins capture both static characteristics and dynamic behaviors of their real-world entities. Based on the glossary of the Digital Twin Consortium, a digital twin may have a stored representation and a computational

[99] Some are more comfortable to consider such synchronized-only-once digital twin as a *model* of the real-world asset in contrasting to and emphasizing the aspect that a digital twin `lives and breathes` with its real-world counterpart. This report does not take a position over this stronger requirement on digital twin but does point out the different considerations.

representation. The stored representation is structured information representing the characteristics as attributes and states of entities and processes. A computational representation is an algorithm representing processes with inputs, outputs, states and attributes that vary in time. The digital twin may be synchronized with the real entity over its states at a frequency adequate for the specific application.

Digital twin as a set of technologies: Digital twins are a set of enabling technologies with core elements built on connectivity and communication, big data and analytics including machine learning (ML), artificial intelligence (AI), and various (3D, Virtual Reality (VR), augmented reality (AR), physical and system simulation modeling technologies, all being put together to support specific industrial-domain applications ensured by trustworthiness.

All in all, the digital twin paradigm enables the creation of a highly flexible digital environment reflecting the reality we seek to manage and operate optimally. In such a digital environment, new components can be easily integrated, configured and adapted as needed systematically. This capability is essential for achieving interoperability between digital twin components and their real-world counterparts effectively and sustainably. Digital twins as digital representations of the real-world entities can then become a pillar of digital transformation not only within an organization but also in an ecosystem of collaborating companies.

REFERENCES

[AAS-2020] Details of the asset administration shell – Part 1: The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC012.0), German Federal Ministry for Economic Affairs and Energy (BMWi), November 2020,

https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html

[ANSI/ISA-95] Enterprise-Control system integration - Part 1: Models and terminology, IEC 62264-1 Mod, *https://www.isa.org/products/ansi-isa-95-00-01-2010-iec-62264-1-mod-enterprise*.

[AutomationML-2021] Standard data exchange in the engineering process of production systems, 2021,*https://www.automationml.org/wp-content/uploads/2021/06/AutomationML-Brochure.pdf.*

[ASME-1989] Digital representation for communication of product definition data, 1989, *https://standards.globalspec.com/std/437642/ASME%20Y14.26M.*

[ASME-2009] Data specification for properties of machining and turning centers, ASME B5.59-2, 2009.

[ASME-2019] Methods for performance evaluation of computer numerically controlled machining centers, 2019, *http://www.leice.com/upfile/201910/2019102565834009.pdf*.

[Booch-2007] Booch, G., et al: Object oriented analysis and design, Third Edition, Addison-Wesley, 2007.

[CAA-1993] Gomaa, H.: Software design methods for concurrent and real-time systems, In: Díaz-Herrera, J.L. (eds) Software Engineering Education, CSEE 1994, Lecture Notes in Computer Science, vol 750. Springer, Berlin, Heidelberg, 1993, *https://doi.org/10.1007/BFb0017625.*

[CMSD-2010] Core Manufacturing Simulation Data (CMSD) standard, 2010, *https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=31457&PortalId=0&TabId=105.*

[DTC-2022a] Glossary of digital twins, Digital Twin Consortium, *https://www.digitaltwinconsortium.org/glossary/glossary.html#digital-twin.*

[DTC-2022b] IIC and DTC White Paper - IoT security maturity model digital twin profile, Industry IoT Consortium, Digital Twin Consortium, 2022, *https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2022/06/SMM-Digital-Twin-Profile-2022-06-20.pdf.*

[DTC-2023] DTC Technical Brief - Platform Stack Architectural Framework: An Introductory Guide, Digital Twin Consortium, 2023, *https://www.digitaltwinconsortium.org/platform-stack-architectural-fram-formework-an-introductory-guide-form/.*

[DTDL-2022] Digital Twin Definition Language (DTDL) for models, *https://learn.microsoft.com/en-us/azure/digital-twins/concepts-models#digital-twin-definition-language-dtdl-for-models.*

[DIN-2018] Reference model for Industrie 4.0 service architectures - Part 1: Basic concepts of an interaction-based architecture, DIN SPEC 16593-1:2018-04, RM-SA, *https://www.beuth.de/de/technische-regel/din-spec-16593-1/287632675.*

[HLA-2010] IEEE standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - federate interface specification, IEEE Computer Society, ISBN 978-0-7381-6247-8, 2010.

[IEC-61360] Standard data element types with associated classification scheme - Part 1: Definitions – principles and methods, 2017, *https://webstore.iec.ch/publication/28560.*

[IEC-61804-3] Devices and integration in enterprise systems - Function Blocks (FB) for process control and Electronic Device Description Language (EDDL) - Part 3: EDDL syntax and semantics, 2020, *https://webstore.iec.ch/publication/60628*.

[IEC-62378] Project IEC 63278-1 ED1 Asset Administration Shell for industrial applications – Part 1: Asset Administration Shell structure *https://www.iec.ch/ords/f?p=103:38:17191709168718::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:1250,23,103536#*

[IEC-62453] Field Device Tool (FDT) interface specification, 2016, *https://www.fdtgroup.org/tag/iec-62453.*

[IEC-62541] OPC Unified Architecture, Parts 1-23, *https://opcfoundation.org/developer-tools/documents/*.

[IEC-62714] Engineering data exchange format for use in industrial automation systems engineering - automation markup language - Part 1: Architecture and general requirements, 2017, *https://webstore.iec.ch/publication/32339.*

[IEC-62832] Industrial process measurement, control and automation - digital factory framework - Part1: General principles, 2020, *https://webstore.iec.ch/publication/65858.*

[IEEE-P2806] System architecture of digital representation for physical objects in factory environments, 2019, *https://standards.ieee.org/project/2806.html.*

[IIC-2016a] IIC Journal of Innovation - Miller, B. and Lin, S.-W.: Industrial internet: towards interoperability and composability, Industry IoT Consortium, June 2016, *https://www.iiconsortium.org/news/joi-articles/2016-June-Industrial-Internet-Towards-Interoperability-and-Composability.pdf*.

[IIC-2016b] IIC Framework - The industrial internet of things Volume G4: Security framework, Industry IoT Consortium, September 2016, *https://www.iiconsortium.org/iisf/*.

[IIC-2020a] IIC White Paper - IIC Digital transformation in industry white paper, Industry IoT Consortium, July 2020, *https://www.iiconsortium.org/pdf/Digital_Transformation_in_Industry_Whitepaper_2020-07-23.pdf.*

[IIC-2020b] IIC White Paper - Digital twins for industrial applications, Version 1.0, Industry IoT Consortium, February 2020, *https://www.iiconsortium.org/pdf/IIC_Digital_Twins_Industrial_Apps_White_Paper_2020-02-18.pdf.*

[IIC-2021a] IIC Framework - The industrial internet of Things trustworthiness framework foundation, Industry IoT Consortium, July 2021, *https://www.iiconsortium.org/pdf/Trustworthiness_Framework_Foundations.pdf*.

[IIC-2021b] IIC White Paper - Characteristics of IIoT information models, Industry IoT Consortium, 2021, *https://www.iiconsortium.org/pdf/Characteristics-of-IIoT-Information-Models.pdf.*

[IIC-2022a] IIC Framework - The industrial internet of things vocabulary, Version 3.0, Industry IoT Consortium, March 2022, *https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/04/Industry-IoT-Vocabulary.pdf*.

[IIC-2022b] IIC Framework – The industrial internet reference architecture, Industry IoT Consortium, 2022, *https://www.iiconsortium.org/iira/*.

[IIC-2022c] IIC White Paper - Identification of information entities, Industry IoT Consortium, August 2022, *https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/08/Identification-of-Information-Entities-2022-08-23.pdf*.

[IIC-2022d] IIC Framework - Industrial IoT artificial intelligence framework, Industry IoT Consortium, February 2022,*https://www.iiconsortium.org/pdf/Industrial-AI-Framework-Final-2022-02-21.pdf.*

[IICP-2020] IIC and Plattform Industrie 4.0 Joint White Paper - Digital twin and asset administration shell concepts and application in the industrial internet and Industrie 4.0, Industry IoT Consortium, Platform Industrie 4.0, October 2020, *https://www.iiconsortium.org/pdf/Digital-Twin-and-Asset-Administration-Shell-Concepts-and-Application-Joint-Whitepaper.pdf.*

[ISO-10303] Industrial automation systems and integration - product data representation and exchange - Part 1: Overview and fundamental principles, 2021, *https://www.iso.org/standard/72237.html.*

[ISO-10303-105] Industrial automation systems and integration - product data representation and exchange - Part 105: Integrated application resource: Kinematics," 2014, *https://www.iso.org/standard/64294.html*.

[ISO-13041] Test conditions for numerically controlled turning machines and turning centers - Part 2: Geometric tests for machines with a vertical work holding spindle, 2020, *https://www.iso.org/standard/73489.html.*

[ISO-13374] Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part 2: Data processing,2007, *https://www.iso.org/standard/36645.html.*

[ISO-13399] Cutting tool data representation and exchange - Part 1: Overview, fundamental principles and general information model, 2006, *https://www.iso.org/standard/36757.html.*

[ISO-13548] Industrial automation systems and integration - parts library - Part 42: Description methodology: methodology for structuring parts families, 2010.

[ISO-14649] Industrial automation systems and integration - physical device control - data model for computerized numerical controllers - Part 201: Machine tool data for cutting processes, 2011, *https://www.iso.org/standard/60042.html.*

[ISO-15531] Industrial automation systems and integration - industrial manufacturing management data - Part 44: Information modelling for shop floor data acquisition, 2017, *https://www.iso.org/standard/71064.html.*

[ISO-16400] Automation systems and integration - equipment behavior catalogues for virtual production system - Part 1: Overview, 2020, *https://www.iso.org/standard/73384.html.*

[ISO-19156] Geographic information – observations and measurements, Open Geospatial Consortium and ISO 19156, 2011, *https://portal.ogc.org/files/?artifact_id=41579*.

[ISO-23247-1] Automation systems and integration - Digital twin framework for manufacturing - Part 1: Overview and general principles, 2021.

[ISO/IEC-17826] Information technology - Cloud Data Management Interface (CDMI), 2016, *https://www.iso.org/standard/70226.html.*

[ISO/IEC-20005] Information technology – sensor networks - services and interfaces supporting collaborative information processing in intelligent sensor networks, 2013, *https://www.iso.org/standard/50952.html#:~:text=ISO%2FIEC%2020005%3A2013%20specifies,common%20service%20interfaces%20to%20CIP*.

[ISO/IEC-20922] Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1, 2016, *https://www.iso.org/standard/69466.html*.

[ISO/IEC-27000] Information technology - security techniques - information security management systems - overview and vocabulary, 2018, *https://www.iso.org/standard/73906.html.*

[ISO/IEC-27033] Information technology - security techniques - network security, 2010, *https://www.iso27001security.com/html/27033.html*.

[ISO/IEC-21823-1] Internet of things (IoT) - interoperability for IoT systems - Part 1: Framework, 2019, *https://www.iso.org/standard/71885.html.*

[ISO/IEC-21823-2] Internet of things (IoT) - interoperability for IoT systems - Part 2: Transport interoperability, 2020, *https://www.iso.org/standard/80986.html.*

[ISO/IEC-21823-3] Internet of things (IoT) - interoperability for IoT systems - Part 3: Semantic interoperability, 2021, *https://www.iso.org/standard/83752.html.*

[ISO/IEC-21823-4] Internet of things (IoT) - interoperability for IoT systems - Part 4: Syntactic interoperability, 2022, *https://www.iso.org/standard/84773.html.*

[ISO/IEC-30141] Internet of Things (IoT) - reference architecture, 2018, *https://www.iso.org/standard/65695.html.*

[ISO/IEC-30147] Information technology – internet of things - methodology for trustworthiness of IoT system/service, 2021, *https://www.iso.org/standard/53267.html.*

[ISO/IEC-30161] Internet of Things (IoT) - requirements of IoT data exchange platform for various IoT services, 2022, *https://www.iso.org/standard/53281.html.*

[ISO/IEC/IEEE-15288] Systems and software engineering - system life cycle processes, 2015, *https://www.iso.org/standard/63711.html.*

[ISO/IEC/IEEE-21450] Information technology - smart transducer interface for sensors and actuators - common functions, communication protocols, and Transducer Electronic Data Sheet (TEDS) formats, 2010, *https://www.iso.org/standard/54356.html*.

[MDPI-2022] Usländer, T., Baumann, M., Boschert, S., Rosen, R., Sauer, O., Stojanovic, L., Wehrstedt, J.C.: Symbiotic evolution of digital twin systems and dataspaces, Automation, 3(3):378-399, 2022, *https://doi.org/10.3390/automation3030020*.

[MTConnect-2022] MTConnect standardizes factory device data, 2022, *https://www.mtconnect.org/.*

[MTConnect-OPC UA -2019] OPC-UA Companion specification for MTConnect, 2019, *https://www.mtconnect.org/opc-ua-companion-specification*.

[OASIS-2006] OASIS Reference model for service oriented architecture 1.0, 2006, *https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf.*

[OG-2009] Navigating the SOA open standards landscape around architecture, The Open Group, *https://www.opengroup.org/soa/source-book/stds/p2.htm*.

[OMG-SysML] OMG systems modeling language 1.4 and SysML v2 RFP, ISO 19514, 2017, *https://www.omgsysml.org/SysML-2.htm*.

[OPC-UA-2017] OPC-UA services specification, *https://opcfoundation.org/developer-tools/specifications-unified-architecture#:~:text=OPC%20Unified%20Architecture%20Specification,more%20secure%20and%20scalable%20solution*.

[PMML-2018] The Predictive Model Markup Language (PMML) 4.3, 2018, *http://dmg.org/pmml/v4-3/GeneralStructure.html.*

[QIF-2020] The quality information framework standard, 2020, *http://qifstandards.org/*.

[Shao-2014] Shao, G., Jain, S., and Shin, J.: Data analytics using simulation for smart manufacturing - proceedings of the Winter Simulation Conference, 2014.

[Tolk-2007] Tolk, A., Diallo, S., and Turnitsa, C.: Applying the levels of conceptual interoperability model in support of integrability, interoperability, and composability for system-of-systems engineering, Systemics, Cybernetics and Informatics, ISSN 1690-4524, Vol. 5, No

AUTHORS AND LEGAL NOTICE

This document is a work product of the Industry IoT Consortium Digital Twin Interoperability Task Group in the Technology Working Group, co-chaired by Shi-Wan Lin (Yo-I Information Technologies, Co. Ltd), Kym Watson (Fraunhofer IOSB), and Gordon Shao (NIST).

Editor: Shi-Wan Lin (Yo-I Information Technologies, Co. Ltd).

Authors: The following persons contributed substantial written content to this document: Shi-Wan Lin (Yo-I Information Technologies, Co. Ltd), Kym Watson (Fraunhofer IOSB), Guodong Shao (NIST), Ljiljana Stojanovic (Fraunhofer IOSB), Bassam Zarkout (IGnPower Inc.).

Contributors: The following persons contributed valuable ideas and feedback that significantly improved the content and quality of this document: Michael Pease (NIST) and Charles Byers (IIC)

Technical Editors: Charles Byers (IIC CTO) and Stephen Mellor (IIC CTO Emeritus) oversaw the process of organizing the contributions of the above authors and contributors into an integrated document.

COPYRIGHT