SINDIT: A Framework for Knowledge Graph-Based Digital Twins in Smart Manufacturing

An Ngoc Lam[1], Gøran Brekke Svaland[1], Miguel Ángel Barcelona[2], Shane Keaveney[3], Wissam Mallouli[4], Luong Nguyen[4], Assia Belbachir[5], Xiang Ma[1], Akhilesh Kumar Srivastava[1], and Ahmed Nabil Belbachir[5]

[1] SINTEF AS, Oslo, Norway first.last@sintef.no
[2] Instituto Tecnológico de Aragón, Zaragoza, Spain mabarcelona@itainnova.es
[3] Croom Medical, Croom, Co. Limerick, Ireland shane@croommedical.com
[4] Montimage, Paris, France first.last@montimage.com
[5] NORCE Norwegian Research Centre, Grimstad, Norway
{assb,nabe}@norceresearch.no

Abstract. Digital twins are revolutionizing smart manufacturing by facilitating real-time monitoring, simulation, and optimization of physical processes. This paper introduces the SINDIT framework, a comprehensive approach tailored for developing knowledge graph-based digital twins. By seamlessly integrating cognitive capabilities, SINDIT enhances decision-making and operational efficiency within manufacturing systems. Central to its architecture is a robust data pipeline, adept at organizing and linking vast amounts of heterogeneous data, thereby enabling advanced data analytics and reasoning.
Case studies from the pilots of the COGNIMAN project underscore the practical utility and benefits of the SINDIT framework. These studies showcase notable enhancements in predictive maintenance, process optimization, and overall productivity. By harnessing the power of knowledge graphs and cognitive capabilities, SINDIT represents a promising avenue for driving innovation and efficiency in smart manufacturing. Through this framework, manufacturers can achieve a higher level of operational insight and agility, leading to improved performance and competitiveness in the market.

Keywords: Digital Twins · Industry 4.0 · Smart Manufacturing.

1 Introduction

The advent of Industry 4.0 has marked the beginning of a new era in smart manufacturing, characterized by the seamless integration of advanced digital technologies and physical production processes [7]. Among these technologies, Digital Twins (DTs) have emerged as a pivotal innovation, providing real-time virtual representation of physical systems [21]. By leveraging DTs, manufacturers can monitor, simulate, and optimize their operations, leading to enhanced productivity, efficiency, and decision-making capabilities [25].

Despite the significant potential of DTs, there is a pressing need for tools that simplify their development and deployment. As defined in [6], a DT software platform includes various components such as IoT connections for communication with the physical twins, data integration, processing and persistence, development of the DT information model, evaluation of DT algorithms (e.g., data-driven analytics), conducting simulations, and visualization of the DT models and analytic results. Existing DT solutions in the market, such as Azure Digital Twins [13], Amazon TwinMaker [1] and ThingWorx [17] are highly customized and proprietary. While open-source solutions such as Thingsboard [23], Asset Administration Shell (AAS) [3], or Eclipse Ditto [4] and Vorto [5] exist, they are tailored to support only specific steps in the development of a DT. An integrated framework that seamlessly brings these components together to enable an end-to-end DT system is still lacking.

To address these limitations, we aim to extend our previous work [25] and present SINDIT (**SIN**TEF **DI**gital **T**win) as an open-source DT framework that is more flexible and maintainable. SINDIT will support continuous development to be applicable to various use cases in the smart manufacturing domain. Our primary objective is to create a modular and scalable framework that simplifies the development and deployment of DTs. The proposed framework will feature:

- **Flexible Data Integration**: Support for a wide variety of data sources and communication protocols, making it easy to connect to different physical systems.
- **Scalable Data Model**: A generic information model that can handle heterogeneous datasets and diverse types of data, ensuring compatibility and extensibility.
- **Knowledge Graph Integration**: The use of knowledge graphs to provide a comprehensive and interconnected representation of the DT, enhancing data integration, querying, and reasoning capabilities.
- **Modular Architecture**: A four-layer architecture (Data, DT Representation, Service and User Interface) that ensures seamless integration of components, enabling end-to-end development of DTs.

The practical application and validation of the SINDIT framework are demonstrated through its implementation in various use cases. For instance, the DT of the Fischertechnik factory showcases real-time monitoring and anomaly detection, while the COGNIMAN project[6] involves pilots in precision machining and additive manufacturing, highlighting significant potential for improvements in decision-making and operational efficiency.

This paper is organized as follows: Section 2 provides background information on the COGNIMAN project and the state-of-the-art development of DT frameworks. Section 3 details the SINDIT framework, including its architecture and implementation. Section 4 discusses the use cases and pilot applications of the framework in smart manufacturing environments. Finally, Section 5 concludes the paper and outlines future work.

[6] https://cogniman.eu/

2 Background

2.1 COGNIMAN - COGNitive Industries for smart MANufacturing

The COGNIMAN project aims to revolutionize manufacturing through the development and deployment of AI-enhanced robotic systems designed for flexibility, modularity, reasoning, and decision-making. The primary objective is to create a scalable, modular solution that enables quick early adoption in various manufacturing processes that have traditionally been difficult to automate. This involves defining a virtual/simulation platform architecture with highly flexible, reconfigurable, and controllable production for digital cognitive manufacturing.

New technologies play a crucial role in COGNIMAN, particularly DTs, AI, and human-in-the-loop systems. DTs are used for parameter optimization prior to deployment, significantly reducing the validation effort. AI technologies are integrated to provide real-time decision-making capabilities and self-adaptiveness to changing conditions. The human-in-the-loop approach ensures that the developed systems are user-friendly and ergonomic, enhancing safety and collaboration between human operators and robots. This combination aims to foster a seamless integration of advanced technologies in manufacturing environments.

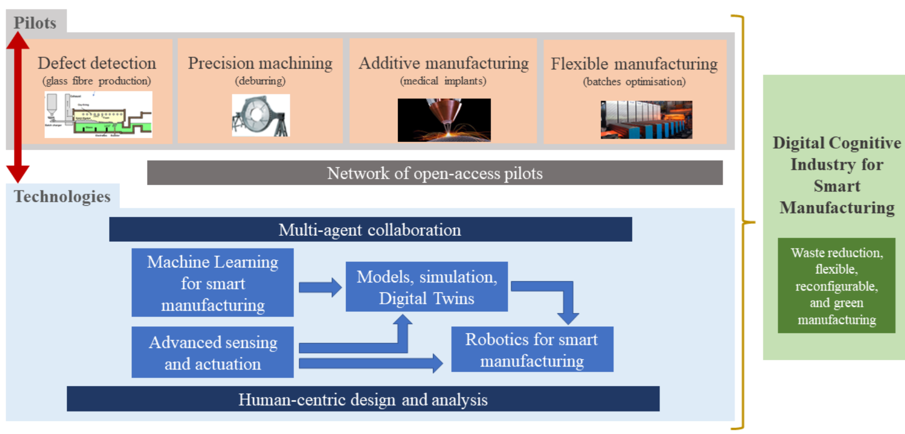

Fig. 1. The COGNIMAN Big picture

COGNIMAN includes four pilot projects to demonstrate the applicability of its technologies. These pilots involve validation in real manufacturing environments to ensure the integrated system works effectively. The pilots cover various manufacturing scenarios: defect detection in fiberglass production, precision machining for deburring large metal parts, additive manufacturing for medical implants, and creating a digital library for batch management in flexible manufacturing. As will be described in the following sections, SINDIT will be specifically used for two of these pilots. The ultimate goal is to create a comprehensive toolbox for smart manufacturing, integrating simulation, models, DTs, AI, sensors, and robotics into a modular solution to automate processes, improve decision-making, and enhance operational efficiency.

2.2 Digital Twin Frameworks

Digital Twins (DTs) are digital representations of physical systems, where physical assets, processes and their relationships are modelled to not only replicate the conditions and status of the physical twin system, but to extend and enhance the system with predictive analysis for future events and simulations of what-if scenarios [25]. There are several actors on the market providing frameworks for building, hosting and managing DT systems. In Table 1, we have listed several proprietary and opensource software systems. The table shows whether the software provides capabilities of remote procedure calls, what data source protocols are supported, and what programming language interface is provided by the software to perform data analytics. On the other hand, in some cases data can be exported to other services that perform the analytics and store the results, that in turn the DT software can consume. A major difference between the proprietary solutions and the opensource solutions are vendor *lock-in* -effects. To make use of the DT service, other services from the same software provider might be required for example to obtain IoT data, perform analytics or to build machine learning models. Opensource frameworks are, in this regard, more customizable, but on the flip side these typically require more of the user to build a fully functioning DT.

Table 1. Overview of various digital twin software solutions. ⌂ ☁ ⇪ : Hosting on-premise, cloud or possibility to upload models to cloud based hosting, respectively.

Software	Pricing model	Open source	Hosting	Remote control	Data sources	Analytics language
Azure Digital Twins [13]	Pay by use	no	☁	✗	Azure IoT services	
Amazon TwinMaker [1]	Tier based pay by use	no	☁	✗	AWS IoT services	
Ansys Twin Builder [2]	Licence based	no	⌂ ⇪	?	?	
ThingWorx [17]	Licence based	no	⌂ ⇪	?	?	
Insights Hub [20]	Licence based	no	⌂ ☁	✓	MQTT, OPC-UA	
ThingsBoard Professional [23]	Tier based	Apache 2.0	⌂ ☁	✓	MQTT, OPC-UA, HTTPS REST, FTP, SNMP, ODBC	Python
ThingsBoard Community	Free	Apache 2.0	⌂	✓	MQTT, OPC-UA, HTTPS REST, FTP, SNMP, ODBC	Python
DTaaS [22]	Free	GPL v3	⌂	?	MQTT, RabbitMQ InfluxDB, MongoDB	
SMOL [8]	Free	BSD-3-Clause	⌂	✗	?	SMOL, FMO
Asset Administration Shell [3]	Free	CC-BY-4	⌂	✓	OPC-UA	
Eclipse Ditto [4] + Eclipse Vorto [5]	Free	EPL v2	⌂	✗	MQTT, AMQP, HTTP, Kafka	
SINDIT [25]	Free	MIT	⌂	✗	MQTT, OPC-UA, InfluxDB	

3 SINDIT Knowledge Graph Based DT Framework

3.1 Software Architecture

SINDIT is structured according to the reference architecture for DT systems developed in COGNIMAN project. It aims to enhance flexibility and modularity through interfaces to connect different components for the purpose of building knowledge graph based DTs. Figure 2 depicts the four-layer architecture of SINDIT along with its constituent components.

Physical Twin The Physical Twin refers to real-world physical assets that are replicated and modeled as DTs within virtual environments. Physical twins can be tangible objects such as sensors, actuators, machinery, or equipment which are monitored during the production in the factory. They can also include intangible artifacts such as processes that need optimization or software systems used in operations. They can even extend to human DTs [24], which represent individuals in virtual form, facilitating personalized simulations, health monitoring, and performance optimization of the operators in the manufacturing environment.

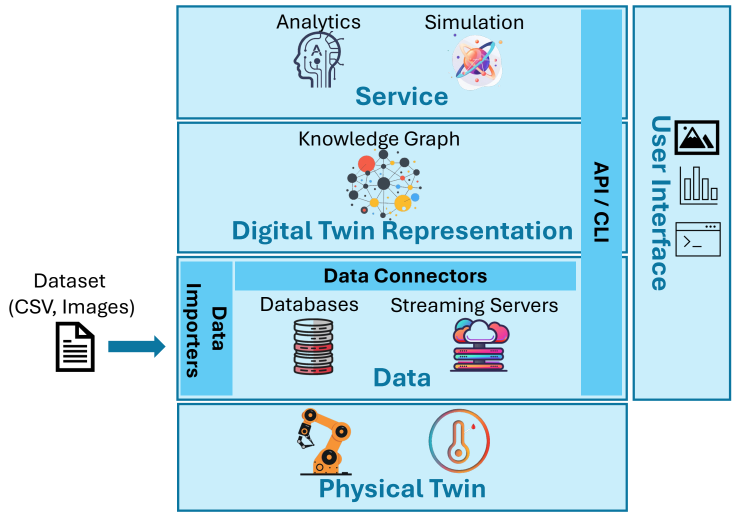

Fig. 2. SINDIT Software Architecture.

Data Layer To interact with DTs, physical twins need to provide interfaces for the collection of data necessary for building their virtual representations and for receiving the control feedback from the DT. This can be achieved by using *data streaming servers* that support bidirectional communication, such as OPC-UA [11], MQTT [18], or RESTful [12] APIs. The Data Layer contains the software components that handle interaction with the physical twins, supporting the collection, storage, and management of data generated by them. It also includes various *databases* to persist the generated data. Additionally, for physical systems that can only export historical datasets manually, the Data Layer provides *Data Importers* to onboard these datasets into its internal databases. The *Data Connectors* are the critical components to make data available to the higher layer. Data Connectors provide a standardized interface to access the data, regardless of the underlying databases and streaming servers. Furthermore, new data connectors can be dynamically registered to support new communication protocols or new database systems employed by the physical twins. Similarly, whenever physical systems generate datasets with a new data format, a new implementation of the corresponding data importer can be added to the Data Layer. This solution enables loose coupling between the layers and among the components, thereby enhancing the modularity and flexibility of the framework.

Digital Twin Representation Layer The Digital Twin Representation Layer employs the *Knowledge Graph* as a conceptualization layer, integrating all the components defined in the Data Layer. The knowledge graph incorporates the metadata of the physical twins and their relationships, providing a comprehensive and interconnected representation of the DT. For scalability and performance efficiency, the knowledge graph does not necessarily contain all historical time-series data already persisted in the Data Layer. Instead, it may capture only the latest values and/or aggregated values (e.g., average, minimum, maximum values) for real-time monitoring purpose. Additionally, the knowledge graph provides the necessary information for the higher layer to retrieve detailed data from the Data Layer. Figure 3 depicts an excerpt from the information model for the knowledge graph within the Digital Twin Representation Layer. This model was developed using the Eclipse Semantic Modelling Framework (ESMF)[7], which includes a meta-model and editor tailored for DT modeling. ESMF also provides standardized vocabularies, such as those for units of measure and data types, facilitating semantic interoperability across different DT frameworks. Thus, by employing ESMF for our DT solution, our goal is to ensure a robust and interoperable representation of our DT system.

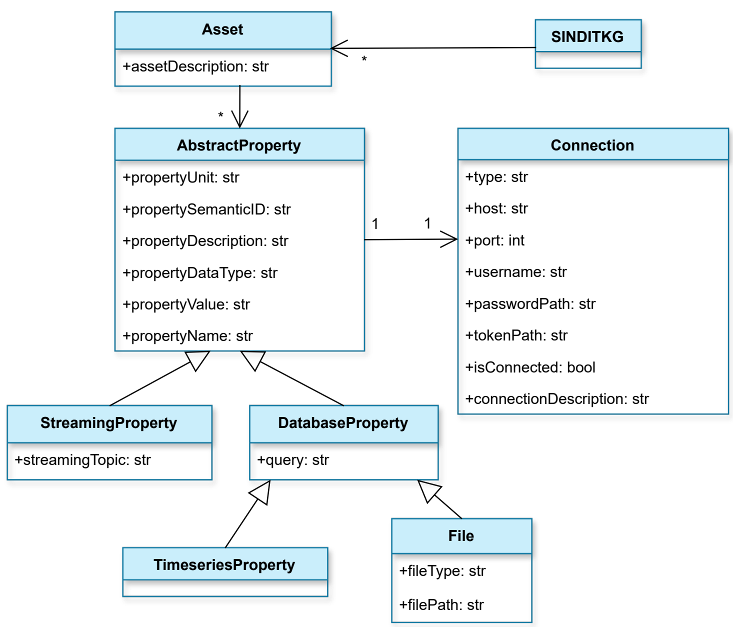

Fig. 3. SINDIT Knowledge Graph Information Model.

As depicted in Figure 3, the *SINDIT Knowledge Graph (SINDITKG)* consists of different *Assets*, which represents the physical devices in the Physical Twins. Each asset may have different *Abstract Properties*, which can be either quantitative (with *Unit* and *Data Type*) or qualitative values. The *Semantic ID* attribute of these properties refers to an externally defined standardized vocabu-

[7] https://eclipse-esmf.github.io/

lary that explains the precise semantic interpretation of the data value. Examples of such vocabularies include the IEC 61360 - Common Data Dictionary[8] for electric/electronic devices and ECLASS[9] for data across various industrial domains. The Semantic ID can also reference a concept defined in other domain-specific OWL ontologies, ensuring comprehensive semantic interoperability within the framework. The *Property Value* attribute maintains the observed value of the property. As previously mentioned, this could represent either the latest observation or accumulated values derived from time-series data.

The *Connection* class within the information model contains details necessary to establish a connection to the underlying database or streaming server. This connection is utilized for instantiating a new Data Connector to the Data Layer. For security reasons, the knowledge graph does not store any credential information (such as passwords or access tokens); instead, it includes paths to external secret vaults where this sensitive information is securely stored.

In addition to the Abstract Property, the information model also includes several concrete properties. The *Streaming Property* facilitates real-time data retrieval from a streaming server, with the Streaming Topic attribute specifying the identifier of the value on the server (e.g., the topic in an MQTT server or the node path in an OPC-UA server). The *Database Property* represents a data value stored within a database system (e.g., a measurement in a time-series database or a column in a relational table), with a query used to retrieve that specific value from the database. Additionally, the *Timeseries Property* and *File Property* denote a continuous value or an object stored in a time-series database or file system, respectively.

Service Layer The Service Layer comprises the functionalities that leverage the data and representation in the Digital Twin Representation and Data Layers to provides various data-driven services. This layer is structured on modular principles, allowing different components to be integrated using standardized APIs provided by the underlying Digital Twin Representation layer. Examples of these services can include:

- Analytics: Performs data analysis to derive insights and predictions.
- Simulation: Conducts simulations to predict outcomes and test scenarios.
- Graph-based Reasoning: Utilizes logical inference and rule-based systems to make decisions and recommendations based on data from the graph.
- Monitoring: Continuously access and update data from the knowledge graph to detect anomalies or deviations from expected norms.
- Control: Uses insights derived from analytics and simulations to automate actions and optimize operations in real-time.

These capabilities leverage the integrated data from the Digital Twin Representation Layer to enhance decision-making and operational efficiencies across diverse applications and industries.

[8] https://cdd.iec.ch/
[9] https://eclass.eu

User Interface The three layers — Data, Digital Twin Representation, and Service — not only provide APIs and CLI for internal communication but also facilitate seamless integration with external systems and applications. The User Interface Layer leverages these interfaces to enhance user interaction. It utilizes APIs and CLI to offer intuitive interfaces for users to interact with DT functionalities. This includes visualizing real-time data, configuring simulations, evaluating different ML models, accessing analytical insights, modifying the knowledge graph to add new assets, controlling the physical devices, and configuring connections through user-friendly dashboards and controls.

3.2 Implementation and Deployment

The implementation and deployment of the SINDIT framework focus on ensuring modularity, scalability, and ease of integration with various data sources and systems. To facilitate rapid prototyping and deployment, the SINDIT framework is containerized using Docker, enabling different components to be deployed as separate containers. Key components include:

Data Layer This layer contains the components supporting bi-directional interaction with the Physical Twins and persisting the generated data. The current version of SINDIT includes components that support MQTT and OPCUA servers. For data storage, it employs InfluxDB[10] for time series data and MinIO[11] for object data such as documents and images. As mentioned earlier, the Data Layer is designed to be easily extended to support different databases or communication protocols. Accordingly, we proposed a unified interface for each type of communication (e.g., to retrieve the database, to interact with the streaming server) so that the integration of new systems is streamlined. This modular design allows for the seamless addition of new functionalities without disrupting existing operations.

For the secret vault, HashiCorp Vault[12] is employed. However, similar to other types of databases, new technologies for secret storage can also be integrated as long as an adapter that implements the secret vault interface is available. This ensures that sensitive information, such as credentials and private keys, is securely stored and managed.

Digital Twin Representation Layer This layer employs a Knowledge Graph to integrate all components defined in the Data Layer, capturing the metadata of the physical twins and their relationships. GraphDB[13]—an RDF triplestore which has shown decent performance as benchmarked in [10]—is employed to store and manage the knowledge graph data within the Digital Twin Representation Layer. Most RDF triplestores provide a SPARQL endpoint, which is a standardized HTTP protocol to query and update semantic knowledge graphs using SPARQL syntax. Consequently, other triplestores can be easily integrated

[10] https://www.influxdata.com/
[11] https://min.io/
[12] https://www.hashicorp.com/products/vault
[13] https://graphdb.ontotext.com/

into the Digital Twin Representation Layer, enabling flexibility and adaptability of the system. This ensures that the knowledge graph can evolve with emerging technologies and changing requirements, providing a robust foundation for advanced data analytics and reasoning within the SINDIT framework.

Service and User Interface Layers These two layers are currently highly application-specific, tailored to meet the precise needs of various use cases. Accordingly, we have different components and dashboards developed for these use cases as described in [25]. Our immediate focus is to ensure that the framework meets the specific requirements of different manufacturing environments, providing optimal performance and relevance.

Looking ahead, our vision is to transform this layer into a more generic and versatile toolkit. Future developments will include a range of machine learning models, enabling users to easily evaluate and deploy these models. This will provide a modular and flexible suite of tools, allowing users to customize and extend the functionalities of the SINDIT framework.

Moreover, similar to the design strategy in the Data Layer, we plan to develop unified interfaces to integrate various simulation technologies and retrieve results from analytics and simulations seamlessly. This standardization will facilitate smoother integration and interoperability, streamline the development of user interfaces, and allow users to leverage advanced simulation and analytical tools without extensive customization efforts.

The SINDIT open-source prototype, written in Python, can be found on the GitHub repository[14], along with a use case for the Fischertechnik Factory. It is also being adapted for two pilots in the COGNIMAN project, specifically within the domains of precision machining and additive manufacturing. Details about these use cases will be discussed in the next section.

4 Use Cases

In this section, we describe the application of the SINDIT framework. The digital twin of the Fischertechnik factory is already implemented, showcasing real-time monitoring and anomaly detection. Additionally, we outline the vision for applying SINDIT in two COGNIMAN pilots: precision machining and additive manufacturing, aimed at enhancing decision-making and efficiency.

4.1 Digital Twin of the Fischertechnik Factory

This section describes the implementation of the DT for the Fischertechnik factory[15], a fully realized application of SINDIT framework. This training factory simulates Industry 4.0 processes, incorporating a high-bay warehouse, a multi-processing station, a sorting line with color recognition, an environmental station with various sensors and surveillance camera, a delivery and pickup station with

[14] https://github.com/SINTEF-9012/SINDIT
[15] https://www.fischertechnik.de

colour detection and NFC reader, and a vacuum suction gripper robot. These physical components constantly send sensor data to the DT through MQTT and OPC-UA servers. The list of 37 sensor values is described in Table 2.

Table 2. Sensor data collected from the Fischertechnik Factory.

Physical Devices	**Sensor Values**	**Streaming Protocol**
Environment Sensor	14 values for temperature, humidity, air quality and pressure, camera positions and its image	MQTT
Muti-Processing Station (MPO)	2 values for MPO status	OPC-UA
Sorting Line (SLD)	2 values for SLD status	OPC-UA
Highbay Warehouse (HBW)	6 values for HBW status and positions	MQTT
Delivery and Pickup (DPS)	4 values for DPS status	OPC-UA
Robot (VGR)	9 values for VGR status and poistions	OPC-UA

The Data Layer of the Fischertechnik DT includes Data Connectors for both MQTT and OPC-UA to provide real-time data retrieval. Additionally, this layer also incorporates an InfluxDB database to store historical sensor values from these two streaming servers, as well as MinIO to store object files such as images taken from the camera, CAD images, and manuals for the physical devices.

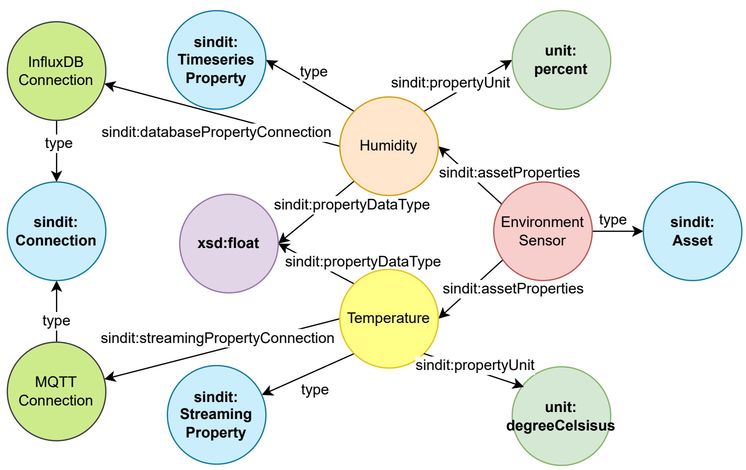

Fig. 4. Snapshot of the SINDIT subgraph for the Fischertechnik factory. Bold nodes are defined in the SINDIT information model; others are the factory instances.

Figure 4 illustrates a subgraph of the Digital Twin Representation Layer where the metadata (e.g., data type, units of measure, connection details) of the humidity and temperature sensor values are captured by the graph. As described in Section 3, the attributes (also known as data properties) of the nodes (e.g., credential information of the connection, data values of the properties) are also captured in the knowledge graph. These attributes can be used to retrieve more data from the Data Layers, such as by instantiating new Data Connectors to the MQTT or InfluxDB servers.

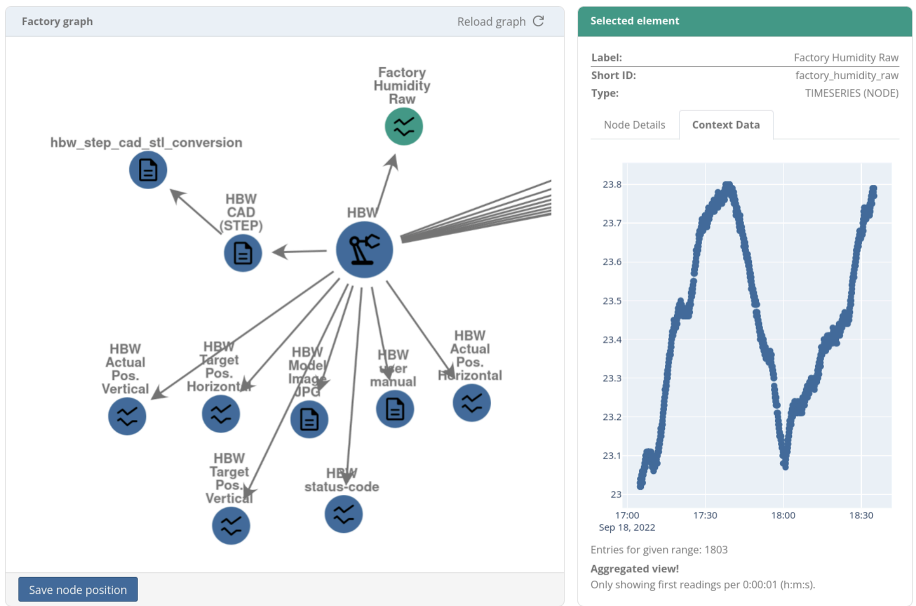

Fig. 5. Fischertechnik Factory Digital Twin Dashboard.

The Digital Twin Representation Layer also provides APIs for higher layers to retrieve data from the knowledge graph, supporting data analytics and visualization. Figure 5 shows a snapshot of the dashboard developed for the DT of the factory. A live version of this user interface is also available online[16]. The dashboard includes a window displaying the full knowledge graph and a side window visualizing the details of the selected node. As shown in Figure 5, the historical data of the humidity value from InfluxDB is visualized in the left sidebar. Furthermore, the Service Layer of the Fischertechnik DT includes a *Similarity Pipeline* for anomaly detection in the time series data with human-in-the-loop interaction. Accordingly, the dashboard allows the user to annotate and label specific time frames of a particular time series data whenever errors occur in the factory (e.g., the sorting line gets stuck, or the NFC cannot read the data). This enables the system to learn from these annotations and improve its detection capabilities over time. By incorporating user feedback, the Similarity Pipeline can more accurately identify anomalies and provide timely alerts to prevent potential issues. This collaborative approach enhances the overall reliability and efficiency of the DT, making it a valuable tool for predictive maintenance and operational optimization. Details about this service can be found at [16].

[16] https://sindit.sintef.cloud/

4.2 Digital Twin Supporting Precision Machining

In this section, we present a pilot case for automating the deburring of large metal parts. We outline the logical blocks to be developed according to the conceptual framework of layers, and we highlight the primary benefits that SINDIT provides in supporting the user stories related to the digital twin.

This pilot aims to create a safe and responsive smart robotic solution for deburring unique and small-batch large metal parts, shifting the physical burden from humans to robots while ensuring high-quality finishes. The specific objectives include: (i) Developing a robot that collaborates with humans, navigates its environment to avoid collisions, and is easy to operate; (ii) equipping the robot with cognitive capabilities through advanced sensors and machine learning to minimize human input, enabling autonomous deburring and quality checks, and allowing the robot to learn from experience for new tasks; and (iii) ensuring the robot can autonomously and safely navigate around the parts.

The development of a series of logical components will be undertaken, detailed according to the COGNIMAN architectural framework as follows: (i) The physical layer includes device-level components such as deburring tool sensors (triaxial accelerometer, force and torque sensor, 2D Gocator laser, touch probe sensor), Automated Guided Vehicle (AGV) sensors (Ridbadge laser 2D), AGV onboard RGB and Light Detection and Ranging (LiDAR) cameras, and external sensors for detecting people and positioning parts; (ii) The data layer comprises datasets like the map and part piece model, deburring trace model, deburring feedback quality model and semantic map model; (iii) The service layer encompasses embedded software tools for navigation and deburring planning, autonomous navigation and deburring, safety awareness, semantic map generation, quality feedback and improvement, and global mission control; (iv) The digital twin layer includes the Gazebo Simulator and a real-time ROS2 board viewer; (v) The UI layer features the Human Machine Interface (HMI), and; (vi) The connectivity layer consists of software bridges for edge-cloud communication using ROS2 Data Distribution Service (DDS) and MQTT protocols.

The robotic solution, as shown in Figure 6, consists of an AGV providing mobility for a robotic arm that performs automatic deburring functions. This system is integrated under a control system, a mobile based user interaction and a external sensor located at the ceiling of the workspace, composed by a LiDAR module and a set of mirrors that concentrate the laser beams to obtain a dense pointcloud of the space below to detect the positions of the robot and the part.

Within the pilot case, several use cases delineate the requirements of various stakeholders regarding the utilization of a DT Users aim to monitor and interact with the robotic solution via a human-machine interface, enabling tasks such as loading maps, parts, and missions, handling alerts, and adjusting priorities. They also seek the capability to perform virtual modifications to assess the robotic deburring performance prior to actual deployment. Additionally, real-time monitoring through a dashboard is desired to understand the solution's operation. For developers, particularly those working on the deburring control and semantic Simultaneous Localization and Mapping (SLAM) generator components, a virtual environment serves as an AI training ground to optimize processes and components. Furthermore, developers aim to specify technical details, including inputs, outputs, configuration parameters, and metadata such as functional descriptions, licenses, and versions, for creating or updating components. They also focus on defining, orchestrating, and parametrizing component compositions at runtime, and ensuring efficient testing and debugging of deployed solutions.

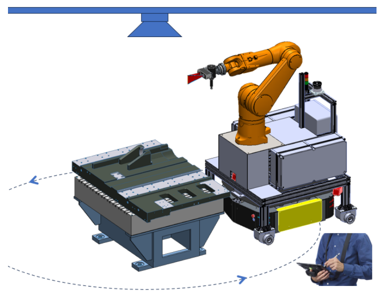

Fig. 6. Conceptual robotic approach for autonomous deburring of large metal parts.

Building upon the needs previously presented, the following lists some of the key features of SINDIT that assist in fulfilling these requirements for creating a DT in the pilot case: (i) The ability to receive information from a robot's physical sensors using various communication protocols. SINDIT includes connectors for MQTT or REST and, through its extensibility to create new plugins or components, a bridge will be integrated to directly support the DDS protocol, connecting SINDIT with a ROS2-based robot at the edge; (ii) The capability to persistently store various datasets, supporting relational, unstructured, or time-series databases. SINDIT includes components to safeguard data in various database management systems and will be able to offer a direct bridge with the COGNIMAN data cloud tool; (iii) The ability to unify and centralize the logs of distributed systems into a single data source, facilitating the monitoring and testing of the solution. SINDIT includes data safeguarding mechanisms and can establish a specific logging source that allows real-time visualization on a dashboard to facilitate debugging; (iv) The capability to include new specific services or user interface widgets for the pilot case that enable autonomous movement and deburring tasks. SINDIT allows extending its service or user interface layers through plugins or its microkernel architecture, so new components can be installed on the platform based on a standard definition and instantiated at runtime to create more complex solutions; (v) The ability to orchestrate business logic through flows that define a composition of services. SINDIT features a design environment where compositions of instantiated services on the platform can be created, along with an engine that supports their interpretation at runtime to build more complex services or automate alert generation; (vi) The capability to have a unified user interface that centralizes all layers of the solution. SINDIT provides a web-based human-machine interface and can be extended with new widgets to include real-time ROS2 dashboards and simulation environment visualizers, offering a single access point to visualize the DT from all perspectives; (vii) The ability to incorporate mechanisms for controlling user permissions and safeguarding SSH aspects. SINDIT includes a user-based authentication layer and, through new services, will encompass SSH-related aspects such as image anonymization, and; (viii) Additionally, for all those advanced functionalities that cannot or should not be integrated into the SINDIT platform itself, a REST API is provided to access each layer programmatically, thus facilitating its extension and coverage for new DT scenarios.

4.3 Digital Twin and Additive Manufacturing

Laser Powder Bed Fusion (LPBF) Additive Manufacturing (AM) processes are widely used in the medical device industry for the creation of complex components due to their high degree of design freedom. The ability of LPBF to create highly complex surfaces and geometries, including metal foams and fine lattice structures, offers great benefits for lightweighting, patient-specific implants, and osteointegration structures, giving it a significant competitive advantage over traditional manufacturing methods. Due to the complex nature of the process and the critical nature of component quality, it is essential that the process is highly qualified and that appropriate controls are put in place. In this use case, the developed framework will be applied to the LPBF process for the collection of meltpool emission data for monitoring and analysis of AM process performance. The goal of this is to develop a DT that can understand process signatures, detect anomalies and failures, and reduce overall production costs. Moreover, a DT of the process will be able to automatically analyze the process monitoring data, detect anomalies or failures, and send alerts during the printing process. It will also be capable of performing more in-depth analysis of the data to generate reports that can compare previous production runs for the same or similar components and output statistical analyses of build performance. This analysis, over time, can be compiled with physical testing and measurements to potentially reduce the quality control checks required for the process.

The 3D printer used in this use case is a Renishaw RenAM500S LPBF system, which is equipped with a single 500W laser (wavelength = 1080 nm). The focused laser spot size diameter is approximately 0.075 mm. The RenAM500S system can 3D print using both continuous and modulated laser modes. For modulated laser processing, the laser fires for a set exposure time at a given power and then switches off, subsequently moving to the next position and repeating the process. The printer's build volume is 250 mm × 250 mm × 350 mm. Example printing conditions used are: layer height: 0.030 mm, power: 200 W, point distance: 0.075 mm, and exposure time: 50 microseconds; however, these vary depending on the area of the component and the type of features to be generated. The Renishaw RenAM500S has process monitoring capabilities built into the machine, specifically the Renishaw InfiniAM Spectral system [9]. This system is equipped with three photodiode sensors that can measure the optical emissions from the laser and the meltpool. To measure the laser emissions, a fixed mirror allows a small amount of laser emission to pass through it and be detected by the LaserView module. The MeltView module has two photodiodes that measure near-infrared plasma emissions in the range of 700 nm to 1040 nm and in the near-infrared range of 1090 nm to 1700 nm. The data acquisition rate for the photodiodes is 100 kHz. The data can be post-processed into a unitless number for each photodiode that represents the emission level, with an associated X and Y coordinate. A TXT file of the raw data is generated per layer, which stores this emission data. An overview of the system is shown in Figure 7.

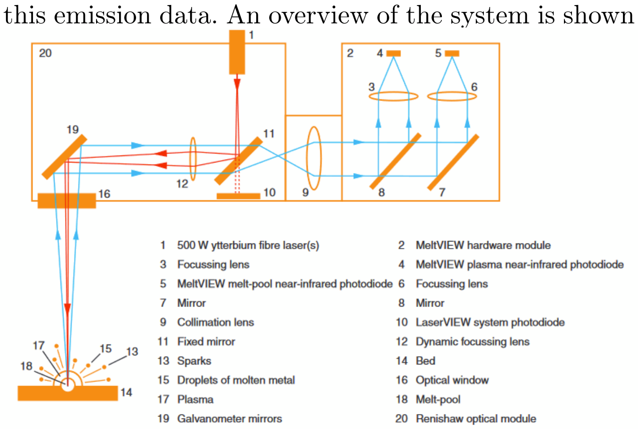

Fig. 7. Schematic of the InfiniAM Spectral system installed on the RenAM 500S LPBF 3D printer [19].

Figure 8 illustrates the overall data flow in the quality control system. The SINDIT system collects printing data from the Renishaw machine, either in batches or as a stream. This data is then preprocessed by SINDIT before being fed into the Digital Twin Model, developed by Montimage. The model evaluates the product printing quality, generating a quality report and providing alerts if any errors or defects are detected. The reports can be sent back to SINDIT for visualizing the quality results, ensuring a more efficient and accurate evaluation of the manufacturing process.

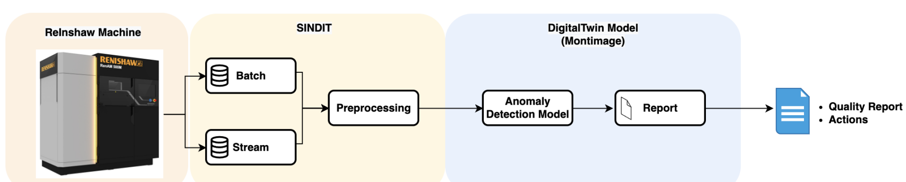

Fig. 8. Data flow of building DT for Additive Manufacturing

4.4 Discussion

The SINDIT modular architecture has demonstrated its flexibility and extensibility in advancing the field of smart manufacturing through the use of knowledge graph-based DTs. By employing a four-layer architecture, SINDIT ensures easy integration of components, facilitating the end-to-end development of DTs. This modularity allows for flexible data integration, supporting a wide variety of data sources and communication protocols. This feature is particularly beneficial in diverse manufacturing environments where different types of machinery and systems need to be interconnected.

The implementation of the DT for the Fischertechnik factory is a prime example of the capability of SINDIT. It showcases the ability of the framework to provide real-time monitoring and anomaly detection through the integration of various sensor data using MQTT and OPC-UA servers. Additionally, the knowledge graph enabled comprehensive information of the physical twins and data querying and reasoning capabilities, which are crucial for predictive maintenance and process optimization of the factory.

Moreover, the COGNIMAN project pilots in precision machining and additive manufacturing further underscore the versatility of the framework in different manufacturing settings. Although not fully implemented, the pilots highlighted the capacity of the framework to support human-machine interaction and interfacing with industrial robots, enhancing decision-making and operational efficiency. In precision machining, the framework facilitated the development of a smart robotic solution for deburring tasks, shifting the physical burden from humans to robots while ensuring high-quality finishes. In additive manufacturing, SINDIT supported the automation of post-processing meltpool emission data for analysis by the developed Digital Twin, reducing costs and improving the quality control monitoring of 3D printed medical implants.

However, several challenges and areas for improvement have been identified. One challenge is the need for advanced security mechanisms which have not been considered in the current implementation. Additionally, more components need to be developed to enhance the functionalities of the framework. For example, the Data Layer needs to be extended with more Data Connectors and Data Importers to support other databases and legacy systems, which are still being used in current industrial settings. A data analytics toolkit for the Service Layer is also required to support the dynamic development and deployment of AI/ML solutions. Furthermore, a new user interface that facilitates the configuration and integration of new components is also needed.

5 Conclusion and Future Work

The SINDIT framework has demonstrated potential in revolutionizing smart manufacturing by providing a comprehensive approach for developing knowledge graph-based DTs. The modular architecture of SINDIT, which includes flexible data integration, a generic data model, knowledge graph integration, and a service-oriented approach, ensures seamless development and deployment of DTs across various manufacturing domains. By providing unified interfaces in each layer, SINDIT facilitates the integration of external solutions for real-time monitoring, simulation, visualization and optimization of physical processes, leading to improved decision-making and operational efficiency.

The practical applications of the SINDIT framework in the COGNIMAN project, particularly in the pilots in precision machining and additive manufacturing and the Fischertechnik factory, underscore its utility and versatility, highlighting the framework's ability to drive innovation and efficiency in smart manufacturing environments.

Looking ahead, several key areas will be the focus of future developments to enhance the SINDIT framework further:

- **Integration with AAS:** Future iterations of SINDIT will incorporate AAS to provide standardized representation and interoperability of DTs, facilitating better data exchange across different platforms and systems.
- **Data Spaces Integration:** Expanding support for Data Spaces [15] will enable comprehensive data sharing, enhancing the scalability and adaptability of DTs solutions in diverse manufacturing settings.
- **Enhanced Machine Learning and Simulation Interfaces:** Developing a suite of machine learning and simulation models within the Service Layer will provide users with modular and flexible tools to customize and extend the functionalities of the SINDIT framework, improving its analytical and predictive capabilities.
- **Advanced Security Orchestration:** Implementing advanced security mechanisms, including explainable security protocols [14], will ensure the resilience and trustworthiness of DT systems against potential cyber threats.
- **Expanded Pilot Applications:** Extending the framework to more pilot projects within the COGNIMAN initiative and other industries will provide insights and validate its effectiveness in diverse manufacturing scenarios.

Acknowledgments. This work has been co-funded by the European Commission Project COGNIMAN (grant agreement No. 101058477) and the SINTEF SEP Project SINDIT 2.0. The work on the Fischertechnik factory was primarily conducted by Timo Peter during his internship at SINTEF, under the supervision of Maryna Waszak.

References

1. Amazon Web Services: Amazon TwinMaker. https://aws.amazon.com/iot-twinmaker/features/, accessed: 2024-06-07
2. Ansys: Ansys twin builder. https://www.ansys.com/products/digital-twin/ansys-twin-builder, accessed: 2024-06-26
3. Bader, S., Barnstedt, E., Bedenbender, H., Berres, B., Billmann, M., Ristin, M.: Details of the asset administration shell-part 1: the exchange of information between partners in the value chain of industrie 4.0 (version 3.0 rc02). Technical report, Federal Ministry for Economic Affairs and Climate Action (BMWK) (2022)
4. Eclipse Foundation: Eclipse ditto. https://eclipse.dev/ditto/, accessed: 2024-06-26
5. Eclipse Foundation: Eclipse vorto (2024), https://eclipse.dev/vorto/, accessed: 2024-06-26
6. Fei, T., Xuemin, S., Cheng, J., Yonghuai, Z., Weiran, L., Yong, W., Hui, X., Tianliang, H., Xiaojun, L., Tingyu, L., et al.: maketwin: A reference architecture for digital twin software platform. Chinese Journal of Aeronautics **37**(1), 1–18 (2024)
7. Ghobakhloo, M.: Industry 4.0, digitization, and opportunities for sustainability. Journal of cleaner production **252**, 119869 (2020)

8. Kamburjan, E., Klungre, V.N., Schlatte, R., Johnsen, E.B., Giese, M.: Programming and debugging with semantically lifted states. In: The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18. pp. 126–142. Springer (2021)
9. Keaveney, S., Shmeliov, A., Nicolosi, V., Dowling, D.P.: Investigation of process by-products during the selective laser melting of ti6al4v powder. Additive Manufacturing **36**, 101514 (2020). https://doi.org/https://doi.org/10.1016/j.addma.2020.101514, https://www.sciencedirect.com/science/article/pii/S2214860420308861
10. Lam, A.N., Elvesæter, B., Martín-Recuerda, F.: Evaluation of a representative selection of sparql query engines using wikidata. In: European Semantic Web Conference. pp. 679–696. Springer (2023)
11. Lam, A.N., Haugen, Ø.: Implementing OPC-UA services for industrial cyber-physical systems in service-oriented architecture. In: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society. vol. 1, pp. 5486–5492. IEEE (2019)
12. Masse, M.: REST API design rulebook: designing consistent RESTful web service interfaces. " O'Reilly Media, Inc." (2011)
13. Microsoft: Azure Digital Twins. https://learn.microsoft.com/en-us/azure/digital-twins/overview, accessed: 2024-06-07
14. Nguyen, T., Lam, A.N., Nguyen, P., Truong, L.: Security orchestration with explainability for digital twins-based smart systems. In: IEEE Annual Computer Software and Applications Conference (2024)
15. Otto, B., ten Hompel, M., Wrobel, S.: Designing data spaces: The ecosystem approach to competitive advantage. Springer Nature (2022)
16. Peter, T.: Introducing Cognition to Digital Twins through Knowledge Graphs and Similarity Measures. Master's thesis, Institut für Software & Systems Engineering, University of Augsburg, Augsburg, Germany (Dec 2022)
17. PTC: PTC ThingWorx IoT Platform. https://www.ptc.com/en/products/thingworx/, accessed: 2024-06-07
18. Quincozes, S., Emilio, T., Kazienko, J.: MQTT protocol: fundamentals, tools and future directions. IEEE Latin America Transactions **17**(09), 1439–1448 (2019)
19. Renishaw: Infiniam spectral (2024), https://www.renishaw.com/resourcecentre/en/details?data=103148, accessed: 2024-08-06
20. Siemens: Insights hub. https://plm.sw.siemens.com/en-US/insights-hub/, accessed: 2024-06-26
21. Soori, M., Arezoo, B., Dastres, R.: Digital twin for smart manufacturing, a review. Sustainable Manufacturing and Service Economics p. 100017 (2023)
22. Talasila, P., Gomes, C., Mikkelsen, P.H., Arboleda, S.G., Kamburjan, E., Larsen, P.G.: Digital twin as a service (dtaas): a platform for digital twin developers and users. In: 2023 IEEE Smart World Congress (SWC). pp. 1–8. IEEE (2023)
23. ThingsBoard Inc.: Thingsboard. https://thingsboard.io/docs/, accessed: 2024-06-07
24. Wang, B., Zhou, H., Li, X., Yang, G., Zheng, P., Song, C., Yuan, Y., Wuest, T., Yang, H., Wang, L.: Human digital twin in the context of industry 5.0. Robotics and Computer-Integrated Manufacturing **85**, 102626 (2024)
25. Waszak, M., Lam, A.N., Hoffmann, V., Elvesæter, B., Mogos, M.F., Roman, D.: Let the asset decide: Digital twins with knowledge graphs. In: 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C). pp. 35–39. IEEE (2022)