Hierarchical Digital Twin Ecosystem for Industrial Manufacturing Scenarios

Matteo Martinelli*, Jingxi Zhang†, Ann-Kathrin Splettstößer†, Marco Picone*, Marco Lippi‡, Andreas Wortmann†
*Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy
†Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW), Germany
‡Department of Information Engineering, University of Florence, Italy

***Abstract*—Modern industrial systems, characterised by distributed and fragmented equipment, present challenges due to their inherent heterogeneity and complexity. This should not impact the stakeholders' business logic, who are more concerned with the information itself rather than how it is collected or processed. Recently, Digital Twins — software copies of physical assets and systems — emerged as a pivotal strategy to bridge the cyber-physical world into an effective digital layer decoupling applications from the management and interaction with physical assets. Fostering this vision, we propose a structured industrial Digital Twins ecosystem exploiting twin relationships and hierarchies to build a digitalised replica of the whole manufacturing system structure enabling a simplified navigation and interaction with the physical world and the data it generates. To support the depicted visions, a fully functioning prototype has been implemented and evaluated in an experimental scenario.**

***Keywords*—Digital Twins, Smart Manufacturing, Cyber-Physical Architecture, Hierarchical Composition**

I. Introduction

Nowadays, industrial production systems face a variety of diverse and heterogeneous challenges, from shorter product life cycles, tighter production budgets, increasing demand for customisation, to the need for energy optimisation to meet environmental goals. As a result, production facilities must be capable of continuously updating to incorporate process improvements and specialised solutions tailored to specific use cases [1]. This requires a high degree of flexibility, but it also makes timely decision-making difficult due to the scattered nature of key information across the shop floor. Extracting valuable insights from this data is time-consuming and complicated by the intrinsic complexity of the systems involved [2]. Moreover, integrating various systems often necessitates multiple handcrafted solutions, which complicates daily operations and makes implementing changes and updates more challenging [3].

The importance of digitalisation in this context cannot be overstated. Digitalising a production system involves mirroring the industrial architecture within a digital environment, capturing critical information such as the relationships between machines, operation orders, and other essential data. This digital reflection is crucial for streamlining operations, enhancing decision-making processes, and maintaining effective production. Digitalisation enables a cohesive and integrated approach, facilitating easier access to relevant information, reducing the complexity of system integration, and allowing for more agile and responsive production environments. By embracing digitalisation, industries can better manage the complexities of modern production, optimise their operations, and stay competitive in a rapidly evolving market.

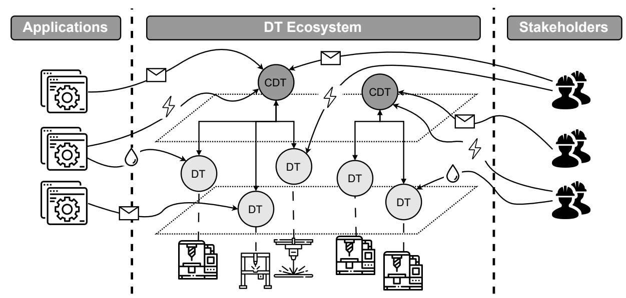


Fig. 1: DT driven industrial cyber-physical ecosystem.

Reflecting an industrial system into the digital environment is challenging: the industrial architecture should be taken into account in digitalising a production system, as it contains vital information, such as relationships between machines, operations orders, or other key information for effective production. Abstracting the real-world structure and having a standardised way to reflect the physical industrial organisation into a composable, digital system, where information is generated, analysed, and made accessible to external applications, can benefit industry managers and related stakeholders. The need for those characteristics lies in existing difficulties in digitalising a complex and heterogeneous environment as an industrial shop-floor [2]. Such difficulties are exacerbated by the dynamic nature of industrial systems, characterised by an inherent propensity to complexity, and by continuous improvement activities as well as product updates, which need adaptations in the associated physical environment over time.

The contribution of this article is to present a Digital Twin (DT) industrial ecosystem organised in a hierarchical architecture featuring data augmentation, actionability, navigability, and composability capabilities [4]. In particular, we: i) proposes a structured DT model for industrial use cases; ii) implement the model in a realistic industrial scenario; iii) evaluate the prototype in a controlled environment; and iv) measure complexity showing the impact of the introduced DT ecosystem and quantifying resource consumption.

The proposed ecosystem acts as a DT-based abstraction layer between the physical world and the digital domain,

2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) | 979-8-3503-8026-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/SEAA64295.2024.00018

2376-9521/24/$31.00 ©2024 IEEE
DOI 10.1109/SEAA64295.2024.00018

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.

organised following the physical world structure, respecting the relationships between physical entities, and collecting and organising data in places where it is logical to obtain them. The architecture has been implemented and tested to assess the benefits of the proposed solution through an industrial distributed heterogeneous and hierarchical system.

II. Related Work

DTs are employed in many different areas of application, e.g., biology, automotive, and manufacturing [5]. In the manufacturing context, a distinction between Digital Models, Digital Shadows and Digital Twins has been made in the literature [6]. In particular, a Digital Model is defined as a representation of a physical object (existing or designed), but without any kind of connection with the physical counterpart. The Digital Shadow has the additional ability to receive data from the physical counterpart but cannot affect it back. Finally, the DT can instead communicate with the physical domain in a bi-directional manner, characterising itself as an active entity with a model of its physical counterpart [7]. Authors in [8] proposed a 5-dimensional model for DTs, evolving the 3D model of Grieves [9]. In particular, the dimensions consider (i) the physical entity, (ii) the virtual model, (iii) the connection between them, (iv) associated data and (v) services.

Practical applications of DTs in manufacturing have been also studied. In [10] a CPS-enabled industrial scenario is equipped with digitalised material handling objects, that are made smart through the use of sensors. The shop-floor state is then reflected in the digital domain using DTs, to track its KPIs,make real-time analyses of them, and take timely decisions, as tighten response logistics activities. Simulation is part of the expected outcome of DTs in manufacturing, and associated capabilities are studied by [11]. In particular, the authors extract a simulation model from production data with modern machine-learning and data-mining tools. The scenario studied involves a quad-copter drone part assembly system, and the simulation goal is to quantify its reliability. The assembly process description is extracted from production data and a Petri-net is computed from it. The extracted Petri-net is then used to analyse the "fail" and "repair" transitions of the system.

In general, DTs are expected to improve the optimisation activities of complex systems as production facilities [5], fostering an advanced application of optimisation models both for industrial operations, as well as for production energy consumption mitigation [12]. DTs application also opens new opportunities on the *product* side, enabling the possibility of interaction between the product and the production system twins. The work in [13] presents a method of product design, manufacturing and services driven by DTs.

Despite the interest in DTs, especially around the concept of industrial digitalisation, most of the present applications are scenario-specific [14]. This denotes a general lack of an approach that *abstracts* each scenario specificity, towards a general method suitable in digitalising an industrial system starting from its base components forming the shop-floor. In this article we explore this area, starting from industrial requirements for digitalisation and contributing to a DT ecosystem that facilitates interaction with upper-level applications.

Twins *aggregations* and *connection* are concepts explored in different ways by the scientific community, to reach interaction among DTs and promote their collaboration. In [15], a six layer architecture for DTs is enriched with aggregation mechanism, with off-the-shelf vendor-neutral software to ensure wider application with heterogeneous hardware settings. Collaboration across different DTs is envisioned also in [16], which approaches the topic under operational, management and security standpoints supported by the proposed architecture.

III. Industrial Requirements

Efficiently managing production requires considering a variety of heterogeneous aspects, including safety, performance, cost tracking, maintenance, quality control, logistics, equipment management, environment, product design and life-cycle, and so on. Stakeholders, as well as industrial managers, all rely on information from the same source, the shop-floor.

Shop-floors are dynamic systems, continuously facing reconfiguration, improvements, and updates. Direct access of upper-level applications towards the complexity captured by IIoT [17] lead to difficulties in uniforming acquired data, selecting information of interest of the given stakeholder and continuously applying value extraction; moreover, travelling the system from higher level applications toward physical actuators to inject actions requests is still hard too, due to the natural equipment heterogeneity of industrial shop-floors. The need for continuous reconfiguration (to adapt the system set-up for the production of different products), improvements (for continuous optimisation of the system under the lean production scope) and updates (to improve the production system capabilities for a new generation of products) clarifies how digitalising, maintaining and updating a physical shop-floor is hard. As systems grow in complexity and market requests change rapidly [2], information and heterogeneity increase, making it difficult to have an organic, real-time representation of the shop-floor state. In the given environment, *uniformly representing* the physical shop-floor neglecting non-trivial details about the equipment gathering the given data, is valuable. In this way each representation of physical assets in the digital domain can be abstracted, promoting the logical scalability of the system. Decoupling between industrial physical assets and associated digital entities is also needed, as it detaches the details of the physical elements concerning its digital representation. Despite being abstraction an interesting point for the digitalisation of industrial systems, the capacity to adapt and personalise the industrial representation is needed, to find a good fit to the specialised needs of each production environment.

Given the need for an industrial digitalised system, it should have a set of features to make it suitable for representing the industrial domain structure in real-time and expose this

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.

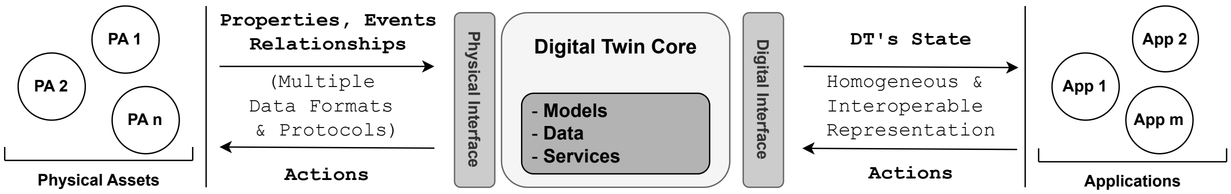


Fig. 2: Abstracted model and structure of a DT with internal components and communication channels.

structure in the digital environment to applications and stakeholders organically, as depicted in Figure 1. Usually, industrial architectures are divided into sub-components, i.e. an entire production plant is divided into departments, and departments are divided again into working areas or production nodes. Production nodes are then constituted by a set of heterogeneous hardware equipment, that could be both automatised in the case of a highly automated node or manually operated. Between department-level production nodes as well as individual production nodes, there are often exploitable relationships; for instance, a robotised production node typically includes input and output buffers, a processing machine, and a robotic arm. Furthermore, components and relationships may evolve due to routine operations or shop-floor upgrades. The abstraction layer must facilitate composability and relationship representation capabilities, enabling the full reflection in the digital domain of possible production system reconfiguration carried out on the physical shop-floor, making it easier to retrieve needed information from the reconfigured system without a full or partial re-design of the digitalisation system.

To enable the described digitalisation of a manufacturing environment, five capabilities (see Figure 3) reported in [18] have been identified and implemented: (i) data ingestion and augmentation, i.e., the ability to ingest data in the digitalised entity that logically follows the state of affairs of the physical world, disregard the protocol used to obtain higher level information; (ii) physical world actionability, i.e., the ability to accept high level actions requests as an input to the digitalised entity, analyse them and pass such requests to associated physical counterparts, monitoring the outcome; (iii) cyber-physical relationships, i.e., the ability to represent any existing relationship between physical objects in their digital counterparts; (iv) composition and hierarchical views, i.e., the ability to compose two or more digitalised entities into one higher level entity, that lives thanks to information flowing from the underlying structures; (v) application interaction, i.e., the ability to offer to the digital domain the information represented by digital counterparts, and collaborating to reach an agreed state of affairs in the real world.

IV. Digital Twin Modelling

Among the different conceptual models for DTs specialised in different target contexts, in this work, very specific characteristics of DTs are required. Since the digitalisation process is expected to start from individual physical objects, which will then be composed into more complex entities, the necessary DT model must be able to represent each physical entity starting from its smallest component. Leveraging explored concepts from existing research [19], proposed models in the manufacturing domain [8], and standardisation bodies [20], the DT model fostered and adopted in this work consists of the following parts (as illustrated in Figure 2): i) *Physical Asset (PA)*: is the digitalised physical entity; ii) *Physical Interface (PI)*: Handles communication with the physical entity, typically using network protocols based on the sensors and actuators involved. Following the 5D model, it represents the *connection* between the physical entity and its virtual counterpart; iii) *Core and Models (M)*: The Core receives data from the PI, processes it based on the best asset model given a set of different models, and potentially computes additional information. The Core stores also the *DT data* gathered by the interaction with the PA, respecting the 5D model; and iv) *Digital Interface (DI)*: Manages interaction with other DTs and applications, exposes services, and grants access to DT data and actuator capabilities. This element corresponds to the *services* dimensions of the 5D model.

Envisioned hierarchies are therefore reached *chaining* different DTs, i.e., connecting the DI of a DT to a PI of a second DT. Since a hierarchy is likely to be composed of several low-level DTs grouped into a higher-level DT, the higher-level DT can accept connections from multiple lower-level DTs through its PI. While the core structure captures asset state, industrial applications require representing relationships between physical equipment (and their DTs). The DT core model incorporates *properties*, *events*, *actions*, and *relationships*. In particular, relationships support hierarchy representation (vertical relationships), model other physical connections (like order between equipment - horizontal relationships), and enable the *navigation* the DT ecosystem without prior knowledge (evaluated in Section VI-B).

The process of digitalising and managing synchronisation between the physical and digital worlds can be denoted as *shadowing* [19]. The DT acts (but is not limited to) as a shadow of the real asset, mirroring its state and capabilities promptly. For accurate DT updates, changes in the physical asset are captured by the PI, and processed by the model, and state changes are shared with the DI. The DT state can only be changed by processing information from the PI. Nevertheless, the DT also accepts action requests from entities external to itself, as other DTs or applications, extending therefore its capabilities from a simple digital shadow. Change requests are introduced into the DT from its DI; then, requests are sent to

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.

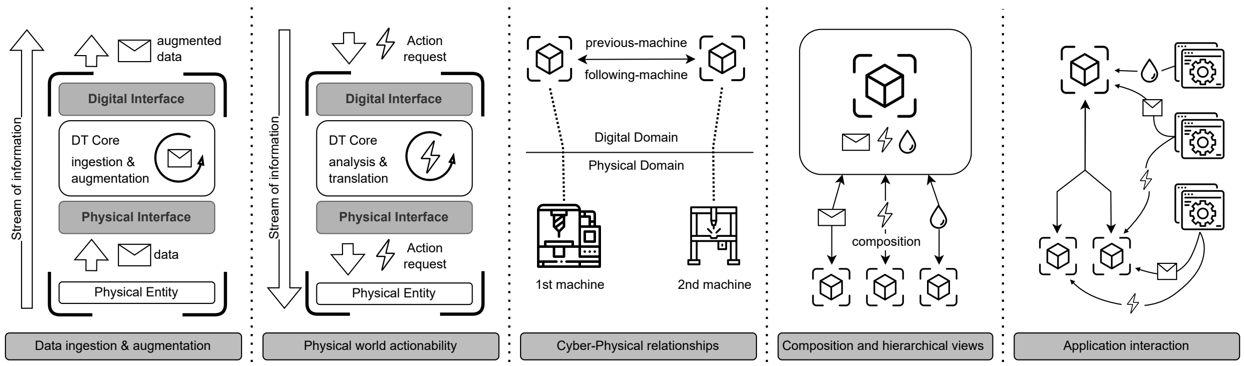


Fig. 3: Vital capabilities of Digital Twins for industrial manufacturing (based on [18]).

the core of the DT where the model *validates* them and then forwards them to the target PA through the PI.

V. Modelling Industrial Digital Twin Ecosystem

In this Section, we present the proposed industrial DTs abstraction focusing on the main modeling patterns to build a structure digital abstraction layer.

A. Data Ingestion & Augmentation

Each machine DT need to have an interface facing the physical world whose responsibility is to ingest information received by the physical world. The interface facing the physical world has to be flexible for the communication needs of each scenario, i.e. should be possible to use different protocols and interaction patterns in the interface to flexibly adapt to the physical world system implementation. A second aspect to consider in the interface is that it doesn't realistically know the information structure received by the physical object. Therefore, is necessary to consider in the interface some description of the physical entity *received* or *built* at the DT start. The description can be grouped into abstracted fields as *properties*, *events*, *actions*, and *relationships* [21]. After getting all machine information, they have to be processed by the DT. Then, information obtained by the physical world has to be passed to the DT core, where they are manipulated following some model or function, and then written as the DT state. Data manipulation is needed to extract some valuable information from the underlying physical entities.

This is the case, for example, of *Overall Equipment Effectiveness* [22], a performance metric for industrial equipment representing how efficient is the system in utilising the production capacity of production equipment in the time domain. OEE can be tracked by a production node DT that monitors its sensors and events and manipulates the received data to understand if the machine is up and producing at the designed speed. After the core manipulation and updates, is exposed externally to other applications or DT entities flexibly concerning the used protocol, through the DT Digital Interface.

B. Physical World Actionability

In the industrial scenario is also needed a pattern of interaction from the digital environment towards the physical world [8]. For example, this can be the case for machine setups, that must follow some specific information usually shared by industrialisation and scheduling offices, maintenance activities, where operators must interact with physical objects and manoeuvre them, or logistics, that must be triggered to coordinate with the production node buffers. Action capabilities have to be exposed by the physical objects through their description as reported in Section V-A, and then are expected to be received from the digital environment (e.g. from another twin or another piece of software used by a system actor). As a consequence, the interaction pattern can be considered as the one depicted for ingesting information in Section V-A, but with the opposite flow: the action request comes from the digital interface, which has the responsibility to correctly handle the request with a suitable communication protocol. After that, the request is ingested by the twin core that eventually analyses it, augments it, or translates it into a set of physical actions. Lastly, the twin core output is given to the physical interface, which has to communicate the result to the underlying physical world.

C. Cyber-Physical Relationships

Physical entities are usually related one to another and between actors in physical world systems. This is also the case for industrial environments, where relationship constraints exist to obtain a certain outcome or logic. The very practical example is represented by industrial layouts, where equipment, production nodes and supportive pillars are grouped and related one to another. A very specific sequence can exist between production nodes, having therefore nodes that come *before* and *after* a given one. Moreover, production nodes can be grouped into clusters (i.e. departments) and clusters can express relations in turn. Operators and equipment can be related to industrial systems, *being part* of one area as well as another. Industrial layouts pose constraints also in KPI and system monitoring: throughput, for example, is a metric that needs to monitor only the *first and last* machines in a grouped system, being thus based on a relationship existing between them. Hence, DTs need to model also relationships between them, concerning the represented physical object. Relationship modelling enables also the possibility of navigating them if the relationship itself stores a pointer to the related digital

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.

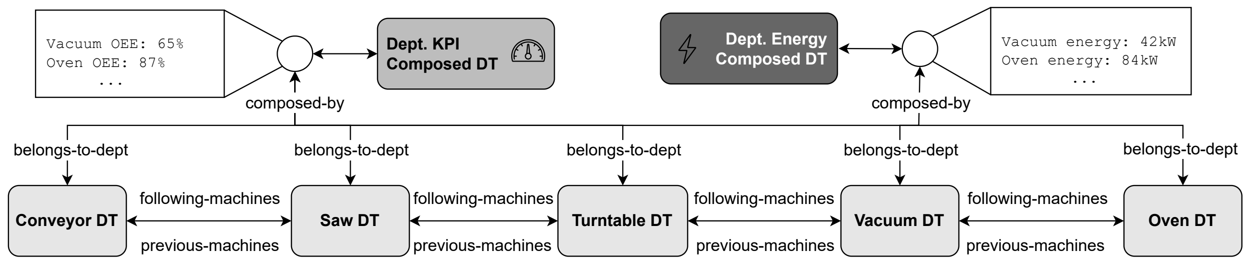


Fig. 4: Compositions and the existing relationships between DTs.

entity. This characteristic is crucial to have the ability for a DT or external software to retrieve the needed information about related DTs.

D. Composition & Hierarchical Views

Industrial architectures are the result of complex composition and interactions between its sub-parts or sub-systems, that need to cooperate to fulfil the common goal of production. Indeed: resources are grouped into departments, departments into business units, and business units into plants. Talking about industrial metrics, a KPI involving composition abstraction is the composed OEE, also called Weighted OEE. Composed OEE is calculated in very different ways [22]. Nevertheless, considering a group of machines going from $i = 1, ..., n$, in the following experiments Weighted OEE is calculated as:

$$\sum_{i=1}^{n} OEE_i \times \frac{net\ available\ time_i}{total\ net\ available\ time} \tag{1}$$

The composition can be also used to create specialised views of the same shop floors, letting the high-level composition ingest only a subset of data from component DTs. As a consequence, a composed DT tracking the department performance through Weighted OEE can exist "in parallel" with another DT composition tracking the overall energy consumption of the same department. This mechanism can be implemented by exposing the digital-side interfaces of a group of DTs to the interface of the physical side of another DT: in this way, DTs exposing their digital-side interfaces act as components, while the DT ingesting data from its physical-side interface is the composed twin. Composed DT interfaces (those facing the physical and the digital side) in this context can be abstracted as input and output interfaces. Flexibility characteristics in terms of communication protocols described in Section V-A are also requested in composed twins as they still need to connect to composition twins and expose information to other applications or twins, without communication constraints and with the best communication patterns. Composition, as depicted so far, is based on a *relationship* between heterogeneous DTs that compose a higher-level entity (e.g., a set of machines composing a department), as reported in Figure 4. Therefore, in setting up a composition, the relation existing between sub-components and the composed DT has to be taken into account when setting the state of all DTs. Recalling Section V-C, relations "*is-part-of*" and "*is-composed-by*" need to be placed in the components and composed DT, respectively.

E. Application Interaction

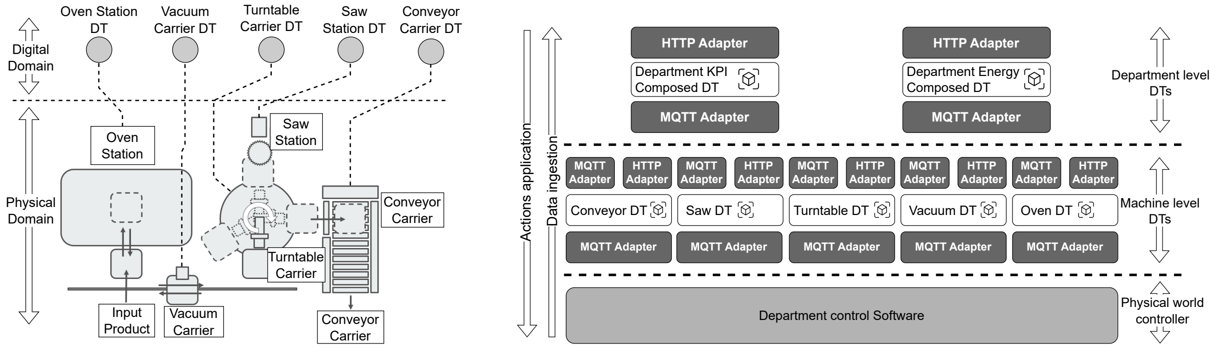
The presented distributed DTs ecosystem represents an effective way to build a digital layer on top of the complexity of the physical world. Thanks to this homogeneous and interoperable level, external applications can interact at different DT levels based on required information, capabilities, and responsibilities. For instance, an app might monitor the entire shop floor, focus on a specific area, or even modify its state of affairs to achieve certain goals. For the OEE status of a department, the external app can query the department's DT. If it seeks the OEE of a specific machine in the composition, information can be obtained from the machine DT or the department's composite DT, depending on the chosen implementation. A similar pattern can hold for action requests. If an action involves a single machine, the external application can make the action request directly to the machine DT. Then, if the DT analyses the request and finds it feasible, it can pass it to the underlying physical object. If an action, instead, involves a group of DTs, the external application can make either one request to each target DT, or one request to the composed DT that, in turn, analyses and shares it with component DTs.

The depicted interaction pattern of interaction can happen when setups are requested at a production changeover. If a setup involves the whole department or a big sub-set of it, e.g. eventuality likely to happen in a cellular manufacturing industrial architecture, is reasonable to have a composed DT for the whole department whose responsibility is also to manage setup requests for each machine. Therefore, the external application interacting to obtain the setup (for example, a scheduling application), will request the department DT. Then, the department DT forwards needed actions to each machine according to its core analysis. If instead, a setup involves a single machine, as can happen in a job-shop layout, where machines are grouped by common working capabilities but raw material usually does not traverse in sequence more than one machine in the same department, a direct setup request is more likely to be passed directly to the machine DT from the external application.

VI. Experimental Evaluation

In the implementation of a practical scenario involving composed DTs, we utilised the *Multi-process Station with Oven*

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 5: *Fischertechnik Multi-process Station with Oven* and first level DTs (a) with their structure and core modules (b).

module from the *Fischertechnik Training Factory Industry 4.0*[1] research equipment, schematised in Figure 5a. This module prototypes a flow shop layout with five machines: 3 material handling machines (a vacuum gripper carrier, a turntable, and a conveyor) and 2 transformation stations (an oven and a saw station). The machines are equipped with light barriers, limit switch sensors, and actuators with 2 or 3 operative states, controlled by a 24V industry-grade digital board. The control hardware consists of two computers: a Raspberry Pi-based soft-PLC directly managing the Fischertechnik factory replica with a 24V Digital Input-Output expansion board and a laptop. These layers are interconnected via an MQTT broker. The envisioned DTs have been implemented using the open-source project WLDT[2], a modular Java software stack designed to effectively implement IoT and IIoT DTs through its communication capabilities, shadowing procedures, and augmentation functionalities [21]. Although other platforms (as presented in [23], [24]) can achieve similar results, the adopted open-source library provided the flexibility to map the envisioned DT modelling, particularly in terms of cyber-physical interactions, modular augmentation, and the digitalisation of cyber-physical relationships with navigability and hierarchical composition.

A. Machine-Layer Digitalisation

For each machine in the production system, a DT has been implemented, categorising the layer as "Machine Layer DT". While it is not mandatory to have a single DT per machine, in this case, a one-to-one mapping has been adopted for simplicity. Each Machine Layer DT is equipped with an MQTT Physical Adapter, and HTTP and MQTT Digital Adapters. The HTTP Adapter exposes endpoints corresponding to the DT nature: the *state* endpoint for retrieving the DT state (read-only), and the *action* endpoint for accepting action requests. The DT state reflects sensor and actuator states, while events, such as *product-on-carrier* and *process-completed*, are tracked under each machine's topic. Augmentation capabilities vary among machines. For the oven DT, real-time power consumption data is received, allowing computation of its energy consumption in the associated Shadowing Function. Other machines also track energy consumption, but based on hypothetical data-sheet power consumption values and events representing the machine *working* state, also mapped in the DTs' Shadowing Function. Based on the machine working state is also OEE, computed equally for all the machines of the use case.

B. Composition & Relationships Navigability

Two composed DTs were created for demonstration: a department-performance composition view and a department-environmental composition view. Composed DTs have an MQTT Physical Adapter subscribed to Machine Layer DT topics, and therefore receiving data from Machine Layer DTs Digital Adapters. The received data is processed through the Shadowing Function and exposed through the Digital Adapter as Machine Layer DTs, in this case, through an HTTP Adapter. In the KPI DT Shadowing Function, composition and augmentation are used to compute the department-weighted OEE and the energy consumption of the entire department. Property updates trigger these calculations, allowing different perspectives of the same production system.

The composed KPI DT includes also throughput computation, a metric representing production rate capability. Through navigability and relationships, the DT tracks when a product starts and ends production in the department, computing throughput via the timestamps received from the first and last machines. In the proposed use case, navigability lies in the fact that the KPI DT, through its composition relationships, *navigates* the structure depicted in Figure 1, retrieving the needed information from components themselves. Then, relationships of retrieved components are analysed, to understand whether the received update comes from the first or last machine in the department. The resulting composition structure, reported in Figure 5b ensures an organic view for stakeholders interested in the production system's performance.

C. Experiments & Results

To assess complexity levels with and without its adoption, we utilised the *Physical Complexity Indicator* (PCI) introduced in [25] to measure the complexity level that an external

[1]Fischertechnik: https://www.fischertechnik.de

[2]White Label Digital Twin - GitHub - https://github.com/wldt

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.

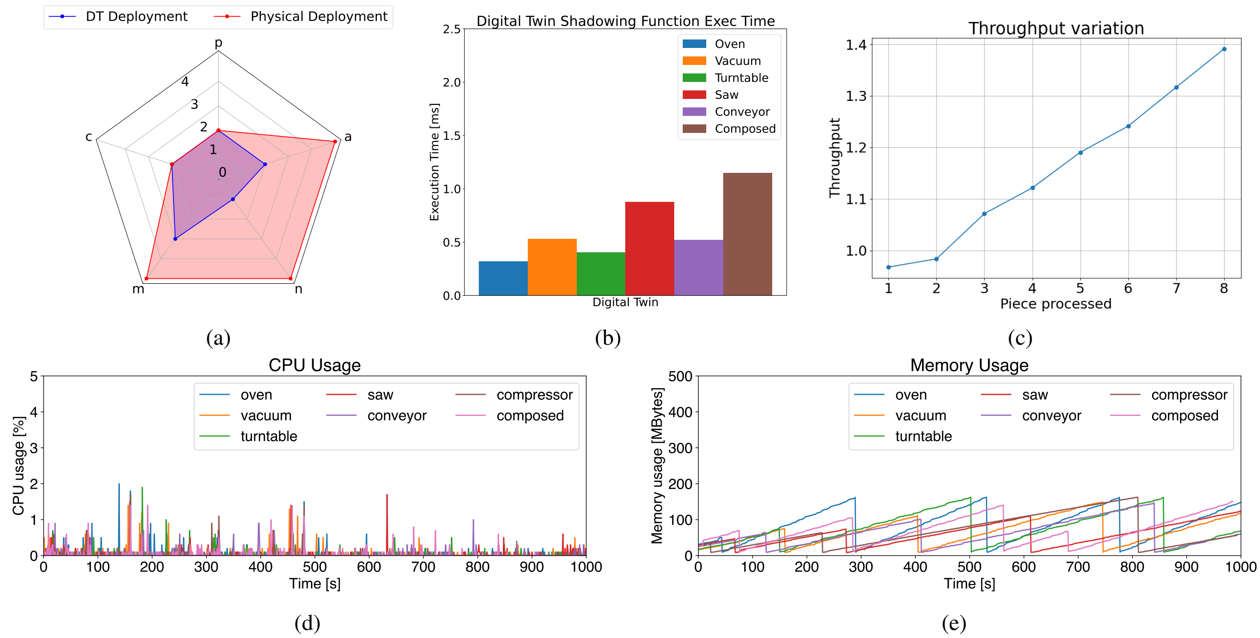


Fig. 6: (a) PCI comparison, (b) Shadowing functions exec. times, (c) throughput variation, CPU (d) and memory (e) load.

Criterion	Imp. Factor	Use Case without DT	Use Case with DT
Required Protocols (p)	2	2	2
Communication Patterns (c)	1	2	2
Data Formats (m)	3	5	3
Interaction Points (n)	2	5	1
Aggregation Points (a)	3	5	2
PCI	–	46	23

TABLE I: Use case's PCI with or without DTs.

application faces to communicate with deployed assets, collect data, and issue commands. PCI is defined considering the following criteria: i) *Required Protocols* (p): the number of application layer protocols needed for digital interaction with deployed physical assets; ii) *Communication Patterns* (c): the number of communication patterns for devices and platforms interaction (e.g., Publisher/Subscriber or Request/Response); iii) *Data Formats*: the variety of data formats, serialisation, and information representation techniques for reading and sending data to deployed devices; iv) *Interaction Points* (n): the number of modules, services, or platforms an application must interact with to retrieve target data or consume services; and v) *Aggregation Points* (a): the levels of aggregation or composition needed to abstract the physical world to the appropriate complexity for observers' typologies and application goals (e.g., merging information and telemetry data from machines in the same production line). Each criterion is assigned an Importance Factor (IF) on a scale from 1 (lower) to 3 (higher) that acts as a multiplier. The index is computed as a weighted sum of the chosen criteria: $PCI = \sum_{i=1}^{5} criterion_i \times IF_i$.

Results are reported in Figure 6a and Table I, showing how DTs standardise data formats, presenting a single interaction and aggregation point to the digital environment. Without DTs, PCI increases with the number of devices and protocols, while DTs decouple physical complexity from the digital aspect, simplifying interaction with external applications. Despite being the proposed case study a simplified scenario, in a real-world layout complexity and fragmentation grow at a high pace. For example, a number of communication protocols higher than 2 are likely to be used, making it hard for higher-level applications to exchange data with the physical layer. With DTs, instead, communication responsibility is delegated to the associated twin, which then translates the communication from the protocol of the machine to the protocol that mostly suites the needs of upper-layer applications. The reduced complexity comes at a computational cost due to the added layer of abstraction. To gauge the impact, CPU, memory, and Shadowing Function execution times were measured during an experiment producing 20 pieces over 17 minutes. Figures 6b, 6d, and 6e show execution times consistently below 5 ms on average, with peak CPU usage at 2%. Memory usage stayed under 200MB due to garbage collection. Specific memory optimisations were not applied to Shadowing Functions, resulting in average execution rates of 0.12 to 1.35 events/second. In an industrial setting, the DT of the whole shop floor can serve as an interface for external applications to monitor and control physical assets. As an example of this functionality, the prototype includes software to dynamically adjust production rates based on market demands or orders. Configuration data is sent to the machine's DT via its HTTP Digital Adapter, as detailed in Section VI-A. The shadowing function of the DT validates the request and forwards it to the physical controller through the physical adapter of the DT. In this case, three machines can adjust their speed: vacuum gripper, turntable, and conveyor

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.

carriers. An experiment was conducted with the following parameters: production of 8 pieces; three-speed settings for each machine (low, medium, high); and an initial low-speed setting. The first two runs maintained the low speed for the oven heat-up phase. From the third to the eighth run, each machine's speed was incremented. DTs observed the real-world state in near-real-time. The Department DT monitored the overall throughput, which was expected to change during the eight runs due to speed adjustments. Figure 6c shows the observed throughput changes, which confirm the behaviour.

VII. Discussion & Conclusion

In this study, we proposed and experimentally evaluated a distributed modelling approach based on DTs to enable and improve industrial digitalisation by creating a distributed hierarchy of twins. The DTs enabled monitoring of production KPIs and energy consumption, with augmentation techniques transforming low-level data into high-level insights. Relationships and system navigability facilitated accurate computation of production throughput, while the DTs accepted actions for real-time adjustments to production rates. Performance assessment included CPU usage, memory utilisation, and execution times of DT shadowing functions. Work limitations involve the usage of industrial protocols as OPC-UA as well as the involvement of industrial stakeholders' feedback: future works will address those aspects, including industrial stakeholders and exploring the use of multiple industrial protocols, focusing on adaptability and scalability of the system. Moreover, future works will include also work-in-process materials scenarios, the applicability to complex industrial environments, and exploring collaborations with external applications to fully harness the potential of DTs in manufacturing.

Acknowledgement

Partially supported by the European Union - NextGenerationEU - under PRIN 2022 project TWINKLE (project ID: 20223N7WCJ, CUP: E53D23007770001). Partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Model-Based DevOps – 505496753. Website: https://mbdo.github.io/

References

[1] N. Kumar, S. Shahzeb Hasan, K. Srivastava, R. Akhtar, R. Kumar Yadav, and V. K. Choubey, "Lean manufacturing techniques and its implementation: A review," *Materials Today: Proceedings*, vol. 64, pp. 1188–1192, 2022, innovative Technologies in Mechanical Engineering-2021 (ITME2021).

[2] G. Herrera Vidal and J. R. Coronado Hernández, "Complexity in manufacturing systems: a literature review," *Production Engineering*, vol. 15, no. 3–4, p. 321–333, Jan 2021.

[3] P. Bellavista, N. Bicocchi, M. Fogli, C. Giannelli, M. Mamei, and M. Picone, "Requirements and design patterns for adaptive, autonomous, and context-aware digital twins in industry 4.0 digital factories," *Computers in Industry*, vol. 149, p. 103918, 2023.

[4] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the iot context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, vol. 108, pp. 1785–1824, 10 2020.

[5] M. Dalibor, N. Jansen, B. Rumpe, D. Schmalzing, L. Wachtmeister, M. Wimmer, and A. Wortmann, "A cross-domain systematic mapping study on software engineering for digital twins," *Journal of Systems and Software*, vol. 193, p. 111361, 2022.

[6] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[7] K. Hribernik, G. Cabri, F. Mandreoli, and G. Mentzas, "Autonomous, context-aware, adaptive digital twins—state of the art and roadmap," *Computers in Industry*, vol. 133, p. 103508, 10 2021.

[8] F. Tao, M. Zhang, Y. Liu, and A. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Annals*, vol. 67, no. 1, pp. 169–172, 2018.

[9] M. Grieves, "Digital twin: manufacturing excellence through virtual factory replication," *White paper*, vol. 1, no. 2014, pp. 1–7, 2014.

[10] W. Wang, Y. Zhang, and R. Y. Zhong, "A proactive material handling method for cps enabled shop-floor," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101849, 2 2020.

[11] J. Friederich, D. P. Francis, S. Lazarova-Molnar, and N. Mohamed, "A framework for data-driven digital twins for smart manufacturing," *Computers in Industry*, vol. 136, 4 2022.

[12] F. Lolli, E. Balugani, M. A. Butturi, A. M. Coruzzolo, A. Ishizaka, S. Marinelli, and V. Romano, "A decision support system for the selection of insulating material in energy retrofit of industrial buildings: A new robust ordinal regression approach," *IEEE Transactions on Engineering Management*, 2022.

[13] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 3563–3576, 2 2018.

[14] Z. Jan, F. Ahamed, W. Mayer, N. Patel, G. Grossmann, M. Stumptner, and A. Kuusk, "Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities," *Expert Systems with Applications*, vol. 216, p. 119456, 4 2023.

[15] A. J. H. Redelinghuys, K. Kruger, and A. Basson, "A six-layer architecture for digital twins with aggregation," in *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, T. Borangiu, D. Trentesaux, P. Leitão, A. Giret Boggino, and V. Botti, Eds. Cham: Springer International Publishing, 2020, pp. 171–182.

[16] P. Kuruppuarachchi, S. Rea, and A. McGibney, "An architecture for composite digital twin enabling collaborative digital ecosystems," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2022, pp. 980–985.

[17] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "A comprehensive survey on interoperability for iiot: Taxonomy, standards, and future directions," *ACM Comput. Surv.*, vol. 55, no. 1, nov 2021.

[18] Digital Twin Consortium, "Introducing an innovative framework for delivering digital twin projects based on use case capabilities," 2022. [Online]. Available: https://www.digitaltwinconsortium.org/initiatives/capabilities-periodic-table/

[19] A. Ricci, A. Croatti, S. Mariani, S. Montagna, and M. Picone, "Web of digital twins," *ACM Trans. Internet Technol.*, vol. 22, no. 4, nov 2022.

[20] ETSI Specialist Task Forces (STF) 628, "Smartm2m; digital twins communication requirements," Tech. Rep. TS-103-845-V1.1.1, February 2024.

[21] M. Picone, M. Mamei, and F. Zambonelli, "Wldt: A general purpose library to build iot digital twins," *SoftwareX*, vol. 13, p. 100661, 1 2021.

[22] R. Gamberini, L. Galloni, F. Lolli, and B. Rimini, "On the analysis of effectiveness in a manufacturing cell: A critical implementation of existing approaches," *Procedia Manufacturing*, vol. 11, pp. 1882–1891, 1 2017.

[23] D. Lehner, J. Pfeiffer, E.-F. Tinsel, M. M. Strljic, S. Sint, M. Vierhauser, A. Wortmann, and M. Wimmer, "Digital twin platforms: Requirements, capabilities, and future prospects," *IEEE Software*, vol. 39, no. 2, pp. 53–61, 2022.

[24] J. Pfeiffer, D. Lehner, A. Wortmann, and M. Wimmer, "Modeling capabilities of digital twin platforms - old wine in new bottles?" *Journal of Object Technology*, vol. 21, no. 3, pp. 3:1–14, Jul. 2022, the 18th European Conference on Modelling Foundations and Applications (ECMFA 2022).

[25] M. Lippi, M. Martinelli, M. Picone, and F. Zambonelli, "Enabling causality learning in smart factories with hierarchical digital twins," *Computers in Industry*, vol. 148, p. 103892, 2023.

Authorized licensed use limited to: Universite de Rennes 1. Downloaded on March 08,2025 at 09:30:22 UTC from IEEE Xplore. Restrictions apply.