Supporting Technical Adaptation and Implementation of Digital Twins in Manufacturing

Enxhi Ferko, Alessio Bucaioni, Moris Behnam
Mälardalen University, Västerås, Sweden
name.surname@mdu.se

***Abstract*—In manufacturing, digital twins are emerging technologies that integrate several advances, such as the industrial internet of things, and cyber-physical systems, for creating software replicas that monitor and control manufacturing units or processes. Despite their great potential and innovation, digital twins are challenging to implement and there is a lack of practical guidelines for their technical adaptation and implementation. This discourages enterprises from planning and adopting full-fledged digital twin-based solutions due to the low return on investment.**

In this paper, we fill such a lack of guidelines for the technical adaptation and implementation of digital twins by providing a catalogue of technologies used for realising digital twin-based systems in manufacturing. We align the catalogue to the International Organization for Standardization 23247 standard for digital twins in manufacturing. We elicit the catalogue by systematically reviewing 14 state-of-the-art DT implementations resulting from a pool of 140 peer-reviewed studies.

To the best of our knowledge, this is the first work that identifies a catalogue of technologies supporting the realisation of digital twin-based systems mapping it into the 23247 standard.

***Index Terms*—Digital twin, manufacturing, ISO 23247, standardisation, architecture, functional entities, development, implementation, tools, technologies.**

I. Introduction

Digital Twins (DTs) are considered to be key enhancers for transforming traditional manufacturing into smart manufacturing. DTs involve the use of digital technologies to create virtual representations of physical processes and assets, enabling smart monitoring, analysis of decisions, and prediction of potentially risky actions [1].

Despite their great potential and innovation, DTs are challenging to implement and current works mostly focus on characterising DT and having a common understanding of what a DT may be and its application [2], [3]. Few works have been discussing the implementation of DTs and technical aspects are yet not well-established: *"no clear and unanimous view on methods and tools to implement DTs into real production environments have been identified [1]"*. The lack of practical guidelines for DTs technical adaptation discourages organisations in manufacturing, especially small and medium enterprises, from planning and adopting full-fledged DT-based solutions [1]. Hughes et al. [4] found that the majority of manufacturing companies declared their expectations of having a one-year Return on Investment (ROI). However, such a goal is unreachable for the reasons mentioned above.

In 2021, the International Organization for Standardization (ISO) released the 23247 standard supporting the development and implementation of DTs in the manufacturing domain. The ISO 23247 standard composes of four parts and includes a DT reference architecture comprising (i) an entity-based reference model and (ii) a functional view [5]. The entity-based reference model divides the DT framework into systems and sub-systems. The functional view identifies, for each of these sub-systems, functionalities that are encapsulated in so-called functional entities (FE). By many, standardisation is seen as a pivotal instrument not only for providing a shared terminology and a framework, but also for describing standardised approaches for the design and implementation of DT systems [6]. However, given the novelty of the ISO 23247 standard, *detailed information on the technical adaptation and implementation of the FEs defined in the reference architecture are currently missing [7].*

In this work, *we tackle the problem of supporting the technical adaptation and implementation of DTs in manufacturing by providing a catalogue of technologies used for realising the functionalities associated with the FEs in the standard.* We elicit the catalogue by systematically reviewing 14 state-of-the-art DT implementations resulting from the set of 140 peer-reviewed studies of our previous mapping study [8]. First, we break down each DT implementation into components that we map to the FEs defined in the standard. Later, we identify the technologies and tools for each of these components.

The contribution of our work is manifold. To the best of our knowledge, this is the first work that identifies a catalogue of technologies supporting DTs realisation in manufacturing and maps it into the ISO 23247 standard. The results of our work fill the gap identified by Lattanzi et al. [1], hence can be used by organisations for achieving one year ROI in DTs [4]. Eventually, our results can help to mature the ISO 23247 standard towards a value-generating tool that improves production processes and operations [9].

The remainder of this paper is as follows. Section II presents an overview of the DT concept and the ISO 23247 standard. Section III details the research methodology to enable independent verification and replication of this research. Section IV presents and discusses the results of our work. Section V gives an overview of related works. Eventually, Section VI concludes the paper with final remarks and future works.

II. Background

In this section, we give an overview of the main concepts we used for this research, DTs and the ISO 23247 standard for DT in manufacturing.

A. Digital Twin

The concept of DT can be traced back to the NASA twinning ideas [10]. In 1970, NASA built a physical replica of the Apollo 13 spacecraft at ground level to test multiple hazardous scenarios [10]. Since then, virtual twins have predominantly replaced physical twins. In the scientific literature, the first formal definition of the DT concept is credited to Grieves and dates back to 2003 [11].

According to Grieves, DT is an information mirroring model of a physical system, which comprises three parts: (i) the physical part, (ii) the virtual model mirroring the physical part, and (iii) the data link allowing bidirectional data flow between the two parts [12]. To date, the DT concept has evolved, and more dimensions have been added to the original concept. A recent survey on the definition of DT in the production domain reports on more than ten different definitions of DTs [13]. Our work adopts the DT definition given in the ISO 23247 standard, which describes *"a digital twin in manufacturing as a fit for purpose digital representation of an observable manufacturing element with synchronisation between the element and its digital representation. A digital twin exists across the entire product life cycle and leverages aspects of the virtual environment (high-fidelity, multi-physics, external data sources, etc.), computational techniques (virtual testing, optimisation, prediction, etc.), and aspects of the physical environment(historical performance, customer feedback, cost, etc.) to improve the performance of the manufacturing system" [5].*

B. ISO 23247

The usefulness and success of the DT heavily depend on employing standardised methods for its implementation. A standard set of building blocks facilitates the extension and reusability of DT systems, while standard interfaces ensure interoperability between different DT systems [9] [14] [15]. ISO23247 provides a framework for developing specific DT implementations. The ISO 23247 goal is to provide guidelines, methods, and best practices to facilitate DT composability and interoperability in the manufacturing domain. The standard consists of four parts. Part 2 defines a *reference architecture* for DTs in manufacturing (Figure 1) and includes (i) an entity-based *reference model* and (ii) a *functional view* of the entity-based reference model with specified FEs.

Each entity has an arbitrary number of sub-entities (grey boxes in Figure 1). The Device Communication entity is divided into two sub-entities and is responsible for collecting data from the Observable Manufacturing Elements (OMEs) and controlling the OMEs. The Digital Twin entity is composed of three sub-entities and is responsible for modelling the data collected from the Device Communication entity and providing functionalities. The User entity uses the services provided by the Digital Twin entity and hosts the application of the framework. The Cross-System entity spans all the other entities to provide common functionalities such as security and data translation assurance. The standard provides a functional view that composes of functional entities (FE) (white boxes

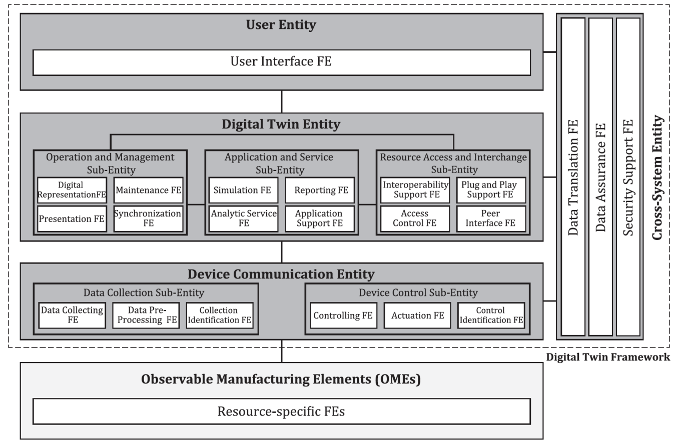

Fig. 1. Functional view of the entity-based digital twin reference model for manufacturing [5].

in Figure 1), refining the entities of the reference model into functionalities to be implemented by DT applications.

In this work, we provide a catalogue of tools and technologies used for implementing DTs in manufacturing and map them to the sub-entities identified by the ISO 23247 standard.

III. Research Methodology

We performed this research using a research methodology that we built on the guidelines for systematic studies in software engineering by Kitchenham et al. [16]. To enable independent verification and replication of this study, we provide a complete and public replication package[1] containing the data from the search and selection, the complete list of primary studies, and the extraction framework. Figure 2 depicts the

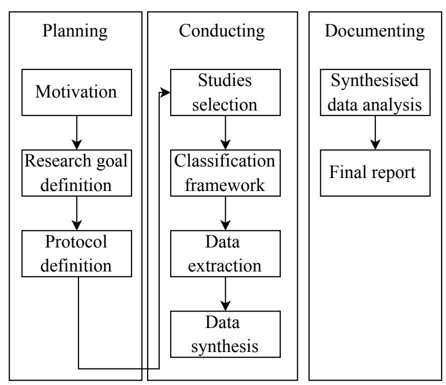

Fig. 2. Research methodology.

research methodology we followed. Our methodology consists of three main phases: *planning*, *conducting* and *documenting*.

In the planning phase, we established the need for this work, identified the research goal (RG), and defined the research protocol to be followed by the researchers to carry out the study in a systematic manner.

In the conducting phase, we identified primary studies starting from the set of 140 studies identified in our previous

[1]The replication package is available at: https://github.com/enxhiferko/ITNG2023

systematic study on DT architectures [8]. To this end, we filtered the 140 studies using the following inclusion criteria (IC):

IC 1 Studies proposing DT architectures targeting the manufacturing domain.

IC 2 Studies describing DT architectures with identifiable and well-documented components.

IC 3 Publication of DT architectures that present concepts or methods related to specific tools and technologies.

Following the filtering process proposed by Ali et al., we only selected those studies that satisfied all the IC [17]. The final set comprised 14 studies, of which 11 described industrial DTs and/or were co-authored by at least one practitioner. We defined the extraction framework that we used to extract the data from the studies. As we wanted to extract the technologies used for implementing DTs in manufacturing and group them per FEs in the ISO 23247 standard, the framework composes of two main parts. The first part contains the list of the 23 functional entities, and domain entities as defined in the ISO 23247 standard. The second part contains, for each of the 23 entities, the actual technology used for implementing the related functionality. We extracted the data from the primary studies using the extraction framework. We thoroughly examined the selected studies to identify the technology used for the implementation and map them to the FEs in the standard. We performed the mapping using the component descriptions in the studies and the entity definitions in the standard. Each author repeated this process independently. In case of uncertainties or disagreements, we added annotations to the respective papers and discussed such annotations in a meeting with all the authors until we reached a consensus. The extracted data are available in the replication package. We used the guidelines by Kasunic et al. [18] and elaborated on the extracted data to provide answers to our research question. We used vertical analysis as it allows for discovering information on each category of the classification framework. For such an analysis, we analysed each primary study individually so to classify its features according to the classification framework. Later, we looked at the whole set of primary studies to reason about potential patterns.

In the documenting step, we wrote this paper, which reports on the results obtained from the data analysis and our observations.

IV. Technical Adaptation and Implementation of Digital Twins

In this section, we present the synthesised data hence the catalogue of technologies realising the functionalities associated with the FEs in the ISO 23247 standard. Table I summarizes the synthesised data. The table is composed of 15 rows of nine columns. Besides the heading row, each row corresponds to one study from the pool of selected studies. The first column specifies the reference to the selected peer-reviewed publication. The second column specifies the OME hence the system for which the DT is designed. Each of the remaining columns maps each of the sub-entities of the ISO 23247 architecture (gray boxes in Figure 1) to the technology or tool used in the study for its technical adaptation and implementation. The mapping between technologies and sub-entities was done using the definitions of the FEs of each sub-entity in the standard and the descriptions of the functionalities provided by the technologies used in the state-of-art implementations. A tool or technology is considered as a realisation for a sub-entity if it covers at least one of its FEs. The OME of the majority of the selected studies is a single piece of equipment in the shop floor, e.g., robot grippers, sorting machines, or a prototype of the entire shop-floor. Only three of the selected studies implemented DT solutions for manufacturing processes. The Data collection sub-entity comprises three FEs, Data collecting, Data pre-processing, and Collection identification. We observed a lack of technology and tools for implementing the Data pre-processing FE. Only a few papers perform data pre-processing by implementing C# scripts [P5], [P16]. The most common way of implementing the Collecting identification FE is by using Radio-frequency identification (RFID) technology [P3], [P16], [P21], [P27]. RFID provides a unique ID and tracks the physical objects throughout the entire lifecycle.

The OPC UA server is the most used technology for implementing the Data collecting FE due to being platform-independent and accommodating multiple communication protocols and different brands of Programmable Logical Controllers (PLCs) [P29]. Very few papers support the technical adaptation of the Device control sub-entity. In these cases, they use PLC [P7], [P18] or commercial IoT, e.g., Thingworx analytics providing for alarm signals, control signals, etc. [P8]. The Operation and management sub-entity consists of four FEs, Digital representation, Maintenance, Presentation, and Synchronisation. The Digital representation FE is mainly realised using standard data models, e.g., Automation ML, Asset Administration Shell (AAS), and Resource Description Framework (RDF), while the Presentation FE is entrusted to CAD software such as Solidworks [P16]. The Maintenance FE is often supported by the ELK stack (i.e., Logstash, Elastic and, Kibana), while more technologies and tools need to support the Syncronisation FE. The Application and service sub-entity consists of four FEs, Simulation, Reporting, Analytic service, and Application support. Current studies mainly support Analytic service and Simulation FEs, which are realised using commercial tools. Some mentioned open-source solutions such as Apache Kafka and Spark. We noticed a lack of technologies and tools implementing the Resource access and interchange sub-entity and its FEs. The User interface FE is mainly implemented using web-based solutions, often within data analysis tools offering dashboard capabilities, too. Among the cross-system entities, most of the current implementations support the technological adaptation of the Security and Data translation FEs.

In the remainder of this section, we describe how each selected study supports the technical adaptation and implementation of sub-entities and FEs of the ISO 23247 standard.

Zheng et al. proposed a generic system architecture for

Paper	OME	Data collection	Device control	Operation and management	Application and service	Resource access and interchange	User Interface	Cross-system entity
[P2]	3D printer	Raspberry Pi	Python scripts, G-Code	Solidwork, Neo 4j	Matlab, Neo 4j	n/a	C# WPF app, .Net framework	OSI model, .STEP format
[P3]	Refinery automation system	RFID, Raspberry Pi V3, modbus protocol	n/a	Automation ML, 3D model in Collada, OpenGL, Node Red	Matlab, Node Red	n/a	Node-Red dashboard	JSON
[P4]	Shop-floor	Maya communication layer, REST/WebSocket	n/a	Kibana, Elasticsearch, Logstash	Maya Simulation framework, Apache Kafka, Apache Spark, Apache Casandra	n/a	Web-based UI	OAuth 2.0
[P5]	Robot gripper	OPC UA server, C# application	n/a	Siemens Tecnomatix Plant Simulation	Siemens Tecnomatix Plant Simulation	n/a	.Net Framework	Hash algorithm, OPC UA
[P6]	Manufacturing processes	Conflict Detection Service, RabbitMQ	n/a	Logstash, Elasticsearch, Kibana	Apache Kafka, Apache Hadoop, TensorFlow	Elasticsearch-Hadoop connector	n/a	n/a
[P7]	Adhesive Melting Machine	EasyModbusTCP	OpenPLC runtime	RDF model	SPARQL queries	n/a	Web interface	RDF format
[P8]	Smart wetland	Thingworx analytics	Thingworx analytics	Thingworx analytics, Vuforia, Rhinoceros	Thingworx analytics	n/a	Vuforia View	n/a
[P15]	Railway axle production line	IoT Gateway, REST API	n/a	Siemens Tecnomatix Plant Simulation	Siemens Tecnomatix Plant Simulation,Python algorithms	n/a	Siemens Tecnomatix Plant Simulation	Encrypted REST API
[P16]	Mock-up shopfloor	OPC UA , NC-Link, RFID tags, C# scripts	n/a	AutomationML, SolidWorks	VisualComponents, VisualField	n/a	Visual Field, WebGL	OPC UA
[P18]	Production process (machine breakdown)	ID number, Raspberry Pi controller, OPC UA	CoDeSys PLC	Simulink	Matlab, Ansys	n/a	Excel	OPC UA
[P19]	Sorting machine	OPC UA server	n/a	Asset Administration Shell metamodel	Apache StreamPipes Runtime	Asset Administration Shell	Apache StreamPipes visual editor	International Data Space connector
[P21]	Mock-up production cell	RFID tags, KepServer	n/a	Flexsim, Node-Red	Flexsim	n/a	Node-Red	n/a
[P27]	Industrial flotation process	RFID tags, RaspberryPi, RESTful interface in Python	n/a	Apache Avro	Apache Kafka, Apache Storm, Apache Spark, Apache Flink, Apache Samza, TensorFlow	n/a	web-based	n/a
[P29]	Welding station	OPC UA server	n/a	Unity 3D, C# scripts	Node.js	n/a	WebGL	JSON, WebSocket, OPC UA

TABLE I
CATALOG OF TECHNOLOGIES UTILISED FROM STATE-OF-ART IMPLEMENTATIONS OF DTS IN MANUFACTURING MAPPED TO THE ISO 23247 STANDARD.

DT establishment in manufacturing [P2]. They validated the proposed architecture by implementing a DT for a 3D printer. A Raspberry Pi microprocessor implement the Data collecting FE. The obtained data is encoded in a standard format according to the OSI model. In control mode, signals are sent to the 3D printer in G-Code format with a back-end Python application. The Presentation FE is realised using a parametric model developed in Solidwork, while the Synchronisation FE is realised with a graph model in Neo 4j. The graph model also supports intelligent decisions of the computation model realised in Matlab and both cover Simulation, and Analytics FE.

Schroeder et al. followed a model-driven approach to support the creation of DT [P3]. To evaluate their methodology, they developed a case study to create the DT of an oil refinery system with four automated valves. They used RFID technology to identify the physical elements uniquely. They modeled the information for the physical devices employing Automation ML and used 3D models in Collada and OpenGL to visualise the components, addressing Digital representation and Presentation FEs, respectively. Node Red is used for monitoring services addressing Maintainability FW, while Matlab to analyse the data realising Analytic FE. User Interface FE is realised with the help of Node-Red dashboard.

Ciavotta et al. proposed a microservice-based platform named MAYA platform, which provides an environment for DT of the entire shop floor and comprises three main components [P4]. MAYA Communication Layer (MCL) covers all the FEs of the Data collection sub-entity. Essentially, MLC is an application runtime environment for distributed automation applications that facilitates aggregation, discovery, orchestration, and seamless communication between physical objects.

MAYA Support Infrastructure (MSI) is another component that makes use of big data technologies such as Apache Kafka, Apache Casandra, and the so-called ELK stack (combination of Elasticsearch, Kibana and Logstash) to monitor, analyse and visualise the data. MSI and MAYA Simulation framework component cover all FEs of both Operation and Management and Application and Service sub-entity. In addition, they have developed appropriate mechanisms for authentication and authorization based on the Oauth2 protocol realising Security FE.

Redelinghuys et al. proposed a six-layer architecture for DT in manufacturing [P5]. A DT for a robot gripper is implemented to evaluate the architecture. They used OPC UA servers and implemented an IoT Gateway in C# to collect and pre-process the data covering Data collecting and Data pre-processing FEs. Tecnomatix Plant Simulation tool is adopted from Siemens, which covers most of the FE of Operation and management, and Application and service sub-entities such as Simulation, Visualization, and Analysis FEs. They implemented several algorithms in C# based on the Hash function to ensure security realising Security FE.

Damjanovic-Behrendt et al. discussed the design and implementation of a DT demonstrator for smart manufacturing [P6]. They chose open-source technology, including data management, modeling, and analytics components. They recommend using RabbitMQ as a messaging protocol that supports several connectivity protocols such as AMQP, MQTT, HTTPS, and WebSocket. RabbitMQ implements the Data collecting FE. Logstash and Elasticsearch are mentioned as powerful tools that handle textual data and are helpful for the analysis of the logs. They support the Digital representation and Maintenance FEs.Kibana is utilised to visually represent the stored log files and customize dashboards, supporting the Presentation FE. Elasticsearch's real-time search and analytic features complement Apache Hadoop's massive data storage and processing power, which support the Analytic FE. Elasticsearch-Hadoop connector is used for their interoperability.

Bamunuarachchi et al. presented a DT prototype implementation for an adhesive melting machine, which supports the Modbus TCP/IP communication protocol [P7]. In order to facilitate interoperability, they recommended a standardised ontology-based approach (RDF model) to model the key hardware and software elements of the machine realising Digital Representation FE and Data Translation FE. They used SPARQL queries to analyse the data covering Analyting FE. OpenPLC runtime realises Controlling and Actuating FEs.

Aheleroff et al. integrated ThingWorx, an industrial IoT platform, Rhinoceros 3D software, and Vuforia, a cloud-based AR software, to develop a DT solution for a smart wetland [P8]. ThingWorx platform covers most of the FEs of Data collection, Operation and management, and Application and service sub-entities such as Data collecting, Presentation, Maintenance, and Analytic FEs, while Vuforia View addresses the User Interface FE.

Riccardo et al. provided an example of how DT is implemented for a railway axle production line [P15]. An IoT gateway was developed to collect data, which was then sent to the cloud via a REST API, realising Data collecting FE. Additionally, Siemens Tecnomatix Plant Simulation was used to build discrete event simulation data and visualize the results addressing Representation, Simulation, Analytics, and User Interface FEs.

Fan et al. presented a general architecture of DT for flexible manufacturing systems [P16]. They developed a proof of concept implementation for a prototype shop floor. Data collecting FE is realised by using universal bus protocols such as NC-Link and OPC UA. C# scripts are developed to support the Data pre-processing FE, while RFID tags are used to target the Collection Identification FE. The Digital Representation FE is realised on AutomationML, while CAD programs such as Solidworks are used to realise the Presentation and Maintenance FEs. They recommended the use of VisualComponents or VisualField to support the Simulation, Analytics, and Reporting FEs. VisualField also supports the User Interface FE.

Barbieri et al. validated their proposed methodology for DT integration into manufacturing systems by implementing DT for reactive scheduling of machine breakdowns [P18]. They utilised a Raspberry Pi controller for data acquisition which covers Data collecting FE. CoDeSys PLC controls and monitors the production process's sensors, addressing Controlling and Actuating FE. They used the SimEvents library from Simulink for data modeling targeting Digital Representation FE, while Matlab and Ansys address Analytic and Simulation FE, respectively. OPC UA is used to build interfaces covering Data translation FE.

Jacoby et al. discussed their experience in realizing a DT architecture based on the ISO 23247 standard with an industrial use case [P19]. They implemented all the FEs of Data collection and device control entity by using an Asset Connection component that reads and writes in a sorting device via the OPC UA server. They built a DT data model based on Asset Administration Shell (AAS) specification. They motivated its use as the specification defines AAS API that is protocol- and technology-agnostic, facilitating interoperability and scalability issues. They employed ApacheStreamPipes, an IoT toolbox that enables data stream analysis for non-technical users, to support all the FEs of Application and Service sub-entity and User interface FE. To support the FEs of the Cross-system sub-entity, they built a component based on International Data Space(IDS) specifications, a data network focusing on data sovereignty.

Pires et al. defined a conceptual architecture for DT that incorporates simulation capabilities to support production process optimization [P21]. As a proof-of-concept, they implemented a DT of a mock-up production cell. Data collecting FE is realised using KEPServer. They integrated KEPServer into their solution because it supports several communication protocols. They developed a discrete-event simulation (DES) model using Flexsim simulation, which supports Digital Presentation FE, and all FEs of Application and service sub-entity. Node-Red for monitoring and visualisation of the results, hence implementing Maintenance FE and User Interface.

López et al. discussed the development of a DT for an industrial floating process using Kafka, an open-source platform for streaming events, analyzing and integrating streaming data [P27]. Their solution supports the Data collecting FE with the help of a Raspberry Pi and Restful interfaces developed in Python. They used RFID tags to identify the data, supporting Collection Identification FE. The Presentation FE is implemented using Apache Avro, an open-source project that provides data serialization, which also enables version management of the DT model. Analytic service and Reporting FEs are realised through Apache Kafka, which is integrated with processing systems such as Apache Storm, Apache Spark, Apache Flink, Apache Samza.

Assad et al. implemented a web-based DT for a welding station [P29]. They utilised the OPC UA server to collect the data as it is interoperable with multiple communication protocols such as Modbus, Profinet, among others covering Data collecting FE. Analytics Fe was realised using a backend application written in Node.js. Results were visualized using Unity3D, a cross-platform game engine, and WebGL, a javascript API for rendering 3D graphics, addressing Presentation and User Interface FE.

V. Related work

To the best of our knowledge, this is the first work that identifies the catalogue of technologies supporting DTs realisation in manufacturing and puts it in relation to the ISO 23247 standard. However, there are previous studies that address complementary aspects.

Dalibor et al. carried out a cross-domain study on software engineering for digital twins [19]. Similar to our work, the authors obtained the results by means of a systematic approach being a mapping study. Among the six research questions driving their work, the third research question focuses on different aspects of engineering DTs, such as processes, tools, etc. Besides having a much wider scope, the main differences with our work are that we explicitly focus on DT implementations in manufacturing and relate the technologies to the functionalities as identified in the ISO 23247 standard.

Parnianifard et al. surveyed the current literature with the aim of reviewing the state-of-the-art and recent developments in DTs [20]. Their review touches upon the most common techniques used for implementing DTs. Here, the main differences in our work are more evident. Parnianifard et al. did not follow any systematic and reproducible approach. Further, they do not provide guidelines for any specific technology or tool nor put them in relation to the standard.

Another systematic review is the one by Botín-Sanabria et al. focusing on defining a comprehensive view of the DT technology and its implementation challenges [21]. Besides looking at different aspects, such as the challenges of implementing DT-based systems using the current technologies, the authors also identify some tools and technologies and cluster them by DT application domains. In light of these premises, the work by Botín-Sanabria et al. can be seen as researching complementary yet important aspects of engineering DTs.

Liu et al. proposed a state-of-the-art survey on digital twin implementations [22]. The survey only focuses on peer-reviewed studies published between 2016 and 2020 and only uses one source indexing system being Google Scholar. In addition, their work focuses mostly on DT definition, enabling technologies, and main functionalities. Interestingly, the authors discussed differences in academic and industrial DT applications. Eventually, in their future direction, they also discuss the need for standardisation efforts.

The work by Lattanzi et al. focuses on analysing different DT ideas and concepts introduced in the literature, posing a special focus on the different methodologies proposed for DTs implementation [1]. Similar to our work, Lattanzi et al. explicitly focus on the manufacturing domain providing a comprehensive description of several DT architectures for the shop-floor. Although they do not focus on technologies, they mentioned some technologies, such as Asset Administration Shell, that they claim may help in standardising the engineering of DTs.

Minerva et al. proposed a survey of technical features, scenarios, and architectural models for DTs in the internet of things context (IoT) [23]. Such a survey is not based on systematic guidelines and mostly focuses on the relationship between IoT platforms and DTs. Among others, the authors characterise the technologies needed for DTs in the IoT context. However, such a characterisation is not detailed and mostly based on common knowledge.

VI. Conclusion and future work

Despite their great potential and innovation, digital twin-based systems are challenging to engineer. To date, only a few works have been discussing how to support the technical adaptation and implementation of digital twins in the manufacturing domain. Such a lack of technical adaptation guidelines discourages manufacturing organizations from planning and adopting full-fledged DT-based solutions [1].

In this work, we tackled the problem of supporting the technical adaptation and implementation of digital twin-based systems in manufacturing by providing a catalogue of technologies used for their realisation. In doing so, we aligned the catalogue to the International Organization for Standardization 23247 standard for digital twins in the manufacturing domain. We elicited the catalogue of technologies and tools by systematically reviewing 14 state-of-the-art digital twin implementations resulting from the set of 140 peer-reviewed studies of our previous mapping study [8]. To the best of our knowledge, this is the first work that identifies a catalogue of technologies supporting DTs realisation in manufacturing mapping it into the ISO 23247 standard. Besides filling the gap for practical guidelines on digital twin realisation, the results of our work can be used by organisations for achieving the desired one year return on investment [4]. Eventually, our results can mature the ISO 23247 standard towards a value-generating tool that improves production processes and operations [9].

Future work may encompass several directions. One direction is to build on the results of this study and investigate how to support interoperability among the tools and technologies identified. Another direction is to use the identified technologies for implementing a digital twin-based system of a manufacturing line as validation of this study.

Acknowledgements

The work in this paper has been supported by the Swedish Knowledge Foundation (KKS) through the ACICS and Modev projects and by the Excellence in Production Research (XPRES) Framework.

Primary studies

[P2] P. Zheng, A. S. Sivabalan, "A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment", *Robotics and Computer-Integrated Manufacturing*, vol.64, pp. 101958, 2020.

[P3] G. N. Schroeder, C. Steinmetz, R. N. Rodrigues, R. V. B. Henriques, A. Rettberg, C. Pereira, "A Methodology for Digital Twin Modeling and Deployment for Industry 4.0", *Proceedings of the IEEE*, vol.109, pp. 556-567, 2020.

[P4] M.Ciavotta, M. Alge, S. Menato, D. Rovere, P. Pedrazzoli, "A Microservice-based Middleware for the Digital Factory", *Procedia Manufacturing*, vol.11, pp. 931-938, 2017.

[P5] A. J. H. Redelinghuys, A. H. Basson, K. Kruger, "A six-layer architecture for the digital twin: a manufacturing case study implementation", *Journal of Intelligent Manufacturing*, vol.6, pp. 1383-1402, 2020.

[P6] V. Damjanovic-Behrendt, W. Behrendt, "An open source approach to the design and implementation of Digital Twins for Smart Manufacturing", *International Journal of Computer Integrated Manufacturing*, vol.32, pp. 366-384, 2019.

[P7] D. Bamunuarachchi, A. Banerjee, P. P. Jayaraman, D. Georgakopoulos, "Cyber Twins Supporting Industry 4.0 Application Development", *International Conference on Advances in Mobile Computing and Multimedia*, pp. 64-73, 2020.

[P8] S. Aheleroff, X. Xu, R. Y. Zhong, Y. Lu, "Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model", *Advanced Engineering Informatics*, vol.47, pp. 101225, 2021.

[P15] I. Ricondo, A. Porto, M. Ugarte, "A digital twin framework for the simulation and optimization of production systems", *Procedia CIRP*, vol.104, pp. 762-767, 2021.

[P16] Y. Fan, J. Yang, J. Chen, P. Hu, X. Wang, J. Xu, B. Zhou, "A digital-twin visualized architecture for Flexible Manufacturing System", *Journal of Manufacturing Systems*, vol.60, pp. 176-201, 2021.

[P18] G. Barbieri, A. Bertuzzi, A. Capriotti, L. Ragazzini, D. Gutierrez, E. Negri, L. Fumagalli, "A virtual commissioning based methodology to integrate digital twins into manufacturing systems", *Production engineering-research and development*, vol.15, pp. 397-412, 2021.

[P19] M. Jacoby, F. Volz, C. Weißenbacher, L. Stojanovic, T. Usländer, "An approach for Industrie 4.0-compliant and data-sovereign Digital Twins Realization of the Industrie 4.0 Asset Administration Shell with a data-sovereignty extension", *At-Automatisierungstechnik*, vol.69, pp. 1051-1061, 2021.

[P21] F. Pires, M. Souza, B. Ahmad, P. Leitão, "Decision Support Based on Digital Twin Simulation: A Case Study", *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pp. 99-110, 2020.

[P25] H. Huang, X. Xu, "Edge computing enhanced digital twins for smart manufacturing", *International Manufacturing Science and Engineering Conference*, 2021.

[P27] C. E. B López, "Real-time event-based platform for the development of digital twin applications", *International Journal of Advanced Manufacturing Technology*, vol.116, pp. 835-845, 2021.

[P29] F. Assad, S. Konstantinov, A. Mus' ab H, E. J. Rushforth, R. Harrison, "Utilising web-based digital twin to promote assembly line sustainability", *IEEE International Conference on Industrial Cyber-Physical Systems*, pp. 381-386, 2021.

References

[1] L. Lattanzi, R. Raffaeli, M. Peruzzini, and M. Pellicciari, "Digital twin for smart manufacturing: A review of concepts towards a practical industrial implementation," *International Journal of Computer Integrated Manufacturing*, vol. 34, no. 6, pp. 567–597, 2021.

[2] P. Zheng and A. S. Sivabalan, "A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment," *Robotics and Computer-Integrated Manufacturing*, vol. 64, p. 101958, 2020.

[3] D. Bamunuarachchi, A. Banerjee, P. P. Jayaraman, and D. Georgakopoulos, "Cyber twins supporting industry 4.0 application development," in *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*, 2020, pp. 64–73.

[4] A. Hughes, "Forging the digital twin in discrete manufacturing, a vision for unity in the virtual and real worlds," *LNS Research e-book*, 2018.

[5] ISO 23247, "Automation systems and integration, digital twin framework for manufacturing," 2021.

[6] A. Lidell, S. Ericson, and A. H. Ng, "The current and future challenges for virtual commissioning and digital twins of production lines," in *10th Swedish Production Symposium (SPS2022)*, 2022.

[7] E. Ferko, A. Bucaioni, P. Pelliccione, and M. Behnam, "Standardisation in digital twin architectures in manufacturing," in *20TH IEEE INTERNATIONAL CONFERENCE ON SOFTWARE ARCHITECTURE*, March 2023. [Online]. Available: http://www.ipr.mdh.se/publications/6600-

[8] "Omitted per double blind reviewing," *Omitted per double blind reviewing*, 2022.

[9] G. Shao and M. Helu, "Framework for a digital twin in manufacturing: Scope and requirements," *Manufacturing Letters*, 2020.

[10] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015, 15th IFAC Symposium onInformation Control Problems inManufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896315003808

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

[12] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.

[13] J. Trauer, S. Schweigert-Recksiek, C. Engel, K. Spreitzer, and M. Zimmermann, "What is a digital twin?–definitions and insights from an industrial case study in technical product development," in *Proceedings of the Design Society: DESIGN Conference*, vol. 1. Cambridge University Press, 2020, pp. 757–766.

[14] G. e. a. Shao, "Use case scenarios for digital twin implementation based on iso 23247," *National Institute of Standards: Gaithersburg, MD, USA*, 2021.

[15] M. Jacoby, F. Volz, C. Weißenbacher, L. Stojanovic, and T. Usländer, "An approach for industrie 4.0-compliant and data-sovereign digital twins," *at-Automatisierungstechnik*, 2021.

[16] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and software technology*, 2013.

[17] N. B. Ali and K. Petersen, "Evaluating strategies for study selection in systematic literature studies," in *Procs of ESEM*, 2014.

[18] M. Kasunic, "Designing an effective survey," Tech. Rep., 2005.

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

[20] A. Parnianifard, S. Jearavongtakul, P. Sasithong, N. Sinpan, S. Poomrittigul, A. Bajpai, P. Vanichchanunt, and L. Wuttisittikulkij, "Digital-twins towards cyber-physical systems: A brief survey," *Engineering Journal*, vol. 26, no. 9, pp. 47–61, 2022.

[21] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sensing*, vol. 14, no. 6, p. 1335, 2022.

[22] Y. Liu, S. Ong, and A. Nee, "State-of-the-art survey on digital twin implementations," *Advances in Manufacturing*, vol. 10, no. 1, pp. 1–23, 2022.

[23] 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, no. 10, pp. 1785–1824, 2020.