Analysing Interoperability in Digital Twin Software Architectures for Manufacturing

Enxhi Ferko[1], Alessio Bucaioni[1], Patrizio Pelliccione[2], and Moris Behnam[1]

[1] Mälardalen University, Västerås, Sweden name.surname@mdu.se
[2] Gran Sasso Science Institute,L'Aquila, Italy patrizio.pelliccione@gssi.it

Abstract. Digital twins involve the integration of advanced information technologies to create software replicas that control and monitor physical assets. Interoperability is an essential requirement in the engineering of digital twins. This paper is the first study analysing interoperability in digital twin software architectures in the manufacturing industry. We began with an initial set of 2403 peer-reviewed publications and after a screening process, we selected a final set of 21 primary studies. We identified the set of technologies used for data exchange and the level of interoperability achieved during such an exchange. We organised the results according to the ISO 23247 standard and the level of conceptual interoperability model.

Keywords: Software Architecture · Interoperability · Digital twin · ISO 23247 · LCIM

1 Introduction

A Digital Twin (DT) is a virtual representation of a physical component, system, or process (i.e., the physical twin) that functions as a digital equivalent for the remote monitoring and controlling of the physical twin [21]. The functional suitability of DTs is heavily dependent on interoperable subsystems that are able to seamlessly and effectively exchange data [28]. Interoperability goes beyond the mere data transmission and is defined as "*the degree to which two or more systems, products or components can exchange information and use the information that has been exchanged*" [18]. Achieving interoperability for DTs can be challenging [20,25]. DTs include a high diversity of subsystems responsible for different functionalities. These subsystems may use different communication technologies for data exchange that are developed without considering the need to operate with each other causing interoperability issues [P5]. This is the case of the ISO 23247 - digital twin framework for manufacturing - standard that provides a functional reference architecture for DTs comprising entities and sub-entities without explicitly discussing how to support interoperability [17]. The observable manufacturing elements (OME) entity and the data collection sub-entity may communicate using a proprietary network with a specialised configuration, while the application service, operation and management sub-entities may use a wired network running IP-based protocols [17]. This diversity in communication technologies makes it challenging to establish interoperability between subsystems, especially when they are developed by different vendors or organisations.

Moreover, as the demand for DT federations increases, achieving interoperability between DTs becomes an upcoming requirement [13]. To the best of our knowledge, we are still missing a comprehensive analysis and assessment of the interoperability requirements and support for DTs.

Therefore, *the research goal (RG) of this paper is to analyse interoperability in DT software architectures for manufacturing*. We analyse how data is exchanged and which level of interoperability is reached during such an exchange in proposed architectures for DTs in manufacturing. We focus on the manufacturing domain for two primary reasons. Firstly, the widespread adoption of DTs in this domain has made it a significant area of interest, with more than 70% of the research on DTs specifically targeting manufacturing [10]. Additionally, the manufacturing domain is only domain that has a dedicated standard for DTs, namely the ISO 23247 standard [17]. We use the ISO 23247 standard together with the Level of Conceptual Interoperability Model (LCIM) [32]. We use LCIM due to its recognition as one of the most effective models for addressing interoperability at early stages of software development, particularly in architectural design [32]. Moreover, LCIM has been successfully applied in several domains, including manufacturing [31]. It is worth noting that adhering to the ISO 23247 standard ensures the broader applicability of our research outcomes [24,13]. We tackled the above goal using a research method built on the guidelines for systematic studies [19]. We analysed 21 DT architectures resulting from a systematic literature review of 2403 peer-reviewed studies. We analysed the final set of 21 DT architectures following a data extraction, analysis, and synthesis process. We identified the technologies employed for data exchange and clustered them according to the network view of the ISO 23247 reference architecture. To indicate the interoperability levels that existed for each of the networks in the proposed DT architectures, we used the descriptive view of the LCIM. In addition, we used the prescriptive view to discuss the requirements necessary to achieve higher interoperability levels.

The remainder of this paper is structured as follows. Section 2 presents an overview of background information. Section 3 describes the adopted research methodology. Section 4 and Section 5 present and discuss the results of this work. Section 6 gives an overview of the related works and Section 7 concludes the paper with final remarks and future works.

2 Background

This section provides an overview of the ISO 23247 standard (part four) and LCIM [32].

2.1 ISO 23247 and information exchange

The ISO 23247 standard comprises four parts [17]. Part four defines the technical requirements for the information exchange between the entities of the reference architecture. In ISO 23247, a network can be seen as a communication point between functional or sub-functional entities of the reference architecture. The

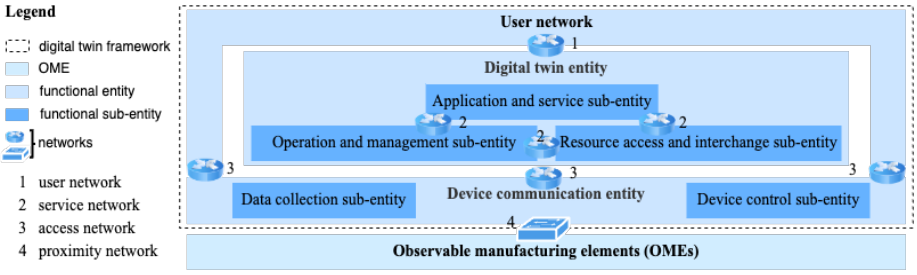

Fig. 1. ISO 23247 networking view of digital twin reference models [17].

standard identifies four types of networks (identified with the numbers 1-4 in Figure 1): user, service, access, and proximity. The proximity network (4 in Figure 1) connects the device communication entity with the OMEs, e.g., resources like equipment. Hence, the device communication entity uses the proximity network, for transmitting commands to and receiving results from OMEs. The access network (3 in Figure 1) connects the device communication entity with the digital twin and the user entity. The digital twin entity synchronises OMEs with their DTs by hosting applications and services such as simulation, analysis, etc. The digital twin entity hosts applications and services using the DT models for humans and other systems. Hence, the device communication entity transmits data collected from the OMEs to the digital twin entity through the data collection sub-entity. Similarly, the device control sub-entity transmits commands from the user entity or the digital twin entity to control the OMEs. The service network (2 in Figure 1) connects digital twin sub-entities among them. Finally, the user network (1 in Figure 1) connects the user entity with the digital twin entity to enable the use of the DT instances managed by the digital twin entity.

2.2 The conceptual interoperability model

A precise understanding of shared data is essential to achieve interoperability between different systems. However, relying solely on standardised data formats such as JSON and communication protocols may not ensure interoperability. According to Carney et al., the assessment of interoperability is crucial and must be measurable to attain success [6]. Several models for evaluating interoperability have been proposed to date. Leal et al. have conducted a thorough review and comparison of 22 such models [22]. In this paper, we refer to the Level of Conceptual Interoperability Model (LCIM) [32]. LCIM identifies 7 levels of interoperability, spanning from no interoperability to conceptual interoperability. At the initial level, level zero, systems function independently and do not share data. At level one, systems can technically exchange data in the form of raw bits and bytes. Moving up to level two, systems use a common data format to achieve syntactic interoperability. However, the meaning of the exchanged data remains undefined at this stage. Semantic interoperability, level three, requires the data meaning to be explicitly specified. At level four, interoperating systems understand the context, system states, and processes, as well as the meaning of the exchanged data, which results in pragmatic interoperability. At level five, dynamic interoperability is achieved as systems can comprehend state changes over

time. Lastly, at level six, conceptual interoperability is attained, where interoperating systems fully comprehend each other's information, processes, contexts, and modelling assumptions.

3 Research methodology

We performed this research using the guidelines for systematic and empirical studies in software engineering [19,29]. Our methodology consists of three phases: planning, conducting, and documenting. In the planning phase, we identified the needs for this study, defined the research goal and questions, and described the research protocol that we followed for carrying out the study. In the conducting phase, we executed all the steps defined in the research protocol, which were search and selection, definition of the classification framework, data extraction and data analysis. In the search and selection step, we exercised the selected scientific databases and indexing systems using the defined search string. We followed a rigorous selection process and filtered the candidate studies to get the final set of primary studies. We complemented the automatic search with fully recursive forward and backward snowballing activities [34]. Using the keywording process [26], we defined a classification framework, and compared and evaluated the primary studies. We used the classification framework to analyse each primary study and extract relevant information through an iterative process. Finally, we analysed the extracted data to answer the elicited research questions. We conducted both quantitative and qualitative analyses. In the documenting phase, we reported on possible threats to validity and related mitigation strategies. To enable independent verification and replication of this study, we provide a complete and public replication package[3] containing the data from the search and selection, data extraction, the complete list of primary studies, and summary of the findings.

3.1 Research goal and questions

Using the Goal-Question-Metric perspectives [4], we defined the RG of this study, that is *(Purpose) Identify, classify, and analyse (Issue) needs, solutions, and challenges of (Object) interoperability in DTs in manufacturing from (Viewpoint) the point of view of researchers.* We broke down the RG in the following research questions (RQs).

RQ1 – How is the data exchanged within a DT and among DTs? We determine which technologies are employed for data exchange. This information is needed for assessing interoperability, identifying its limitations, and assessing potential trade-offs.

RQ2 – Which interoperability levels do current DT implementations reach? We determine the extent of interoperability attained by existing DT implementations according to LCIM [32], together with the identification of challenges that may hinder seamless integration within and across DTs.

[3] The replication package is available at https://anonymous.4open.science/r/analysing-interoperability-replication-package-ECSA2023/README.md

3.2 Search and selection process

Following the steps described in Figure 2, we identified the set of primary studies. We started with the search string (*"Digital Twin" AND Architect**) and queried four of the largest and most reputable scientific databases and indexing systems in software engineering [5,19]: *IEEE Xplore Digital Library*, *ACM Digital Library*, *SCOPUS*, and *Web of Science*. We opted for a concise search string that could help gathering as many relevant studies as possible that we filtered through the application of selection criteria, mitigating potential threats to construct validity. The automatic search on title, abstract and keywords provided

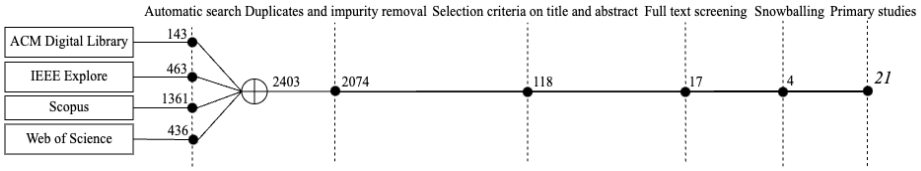

Fig. 2. Overview of the search and selection process

an initial set of 2403 studies, from which we removed impurities and duplicates, and obtained a new set of 2074 studies. Following the recommendations in [1], we applied the following selection criteria to the title, abstract, and keywords, and selected only those studies that satisfied all the Inclusion criteria (IC) and Exclusion criteria (EC). The IC are: (i) studies proposing a DT architecture in the manufacturing domain, (ii) studies proposing DT architectures with well-identified and documented components, (iii) studies providing implementation details on how comprised components of the architecture exchange data, (iv) peer-reviewed studies [33], (v) studies written in English, and (vi) studies available as full-text. The EC are: (i) secondary or tertiary studies, (ii) studies published as tutorial papers, short papers (less than 5 pages), poster papers, editorials and manuals. We obtained a new set of 118 studies, and, by analysing the full text, we selected 17 primary studies. To reduce possible threats to construct validity we performed closed recursive snowballing activities [34]. As a result, we obtained the final set of 21 primary studies that are shown at the end of the paper.

3.3 Classification framework and data extraction

We built a classification framework for extracting and classifying information from the primary studies (Table 1). The framework comprises two facets, one for each RQ. For RQ1, we collected the list of technologies like protocols, stan-

Facet	Category	Description	Value
RQ1	Technologies	Technologies (e.g., protocols, standards, data models) enabling interoperability	String
RQ2	Levels of interoperability	Levels of conceptual interoperability [32] reached	No interoperability, technical, syntactic, semantic, pragmatic, dynamic, conceptual

Table 1. Classification framework

dards, data models, models, etc. used for enabling interoperability. For RQ2, we

collected the levels of conceptual interoperability [32] reached by the solution described in the study. For both RQs, we grouped the collected information according to the network defined in the ISO 23247 standard [17]. We arranged the collected information into groups similar to the sorting phase of the grounded theory methodology [8]. During the data extraction, we refined the classification framework with additional information. Hence, we analysed again the primary studies according to the refined framework and extracted data.

3.4 Data analysis and synthesis

We used the recommendations by Cruzes et al. [9] for analysing and synthesising the extracted data according. We performed vertical analysis for discovering information on each category of the classification framework. In particular, we analysed each study individually and categorised its features using the classification framework. Later, using the line of argument synthesis [33], we reasoned on the entire set of primary studies for uncovering potential patterns.

3.5 Threats to validity

To ensure the *internal validity* of our research, we defined a research protocol using well-established guidelines [19]. Moreover, we employed rigorous descriptive statistical methods for data analysis [35,30] to further mitigate internal validity threats related to data analysis and synthesis. We are confident that the selected primary studies are representative of the population defined by the research questions, as we followed a well-defined and validated protocol. To mitigate threats to *construct validity* associated with data extraction, we developed a framework for extracting data from the studies. Each author independently repeated the process of extracting data from the studies. In case of doubts, the authors added annotations to the respective primary studies and discussed them until reaching a consensus. The ensure *external validity* of our research, we conducted a comprehensive search of four different electronic databases in software engineering and complemented the automatic search with a fully recursive snowballing process. Further, we filtered the studies using selection criteria [1]. We mitigated potential threats to *conclusion validity* by meticulously documenting every step of our research and providing a public replication package to ensure transparency and replicability. In addition, we reduced potential bias during the data extraction process by using well-established models, such as LCIM. All authors participated in data extraction, analysis, and synthesis steps. The conclusions drawn on the interoperability needs and open challenges originated from the primary studies. However, any hypotheses and conjectures were clearly identified as such.

4 Results

We analysed the primary studies and classified their features according to the classification framework in Table 1.

4.1 How is data exchanged? (RQ1)

For each network identified in the ISO 23247 standard, we investigated the primary technologies utilised for data exchange. We focused on two critical aspects of data exchange: data transmission (see Table 2), and data representation

and management (see Table 3). Using the grounded theory methodology [8], we clustered the technologies for data transmission into four groups: protocols, standards, architectural patterns, and open-source platforms. We clustered the technologies for data representation and management in six groups: information models, data formats, graphic APIs, open-source platforms, query language, and standards. This helped us to identify the most commonly utilised technologies and their relationships with one another per each network.

The proximity network enables communication between the device communication entity and OMEs, allowing the device communication entity to receive sensor data from OMEs and send commands to them. In the proximity network, data transmission relies on communication protocols like Profinet and Modbus each of which defines a specific syntax and format. However, there is no apparent consideration for data representation and management within this network. Modbus is the most commonly cited communication protocol in our analysis, appearing in 28% of the primary studies. It is often used to collect data from OME such as sensors, Programmable Logic Controllers (PLCs), and Internet of Things (IoT) devices. In over 50% (11/21) of the primary studies, OME was integrated with the device communication entity within a single system, and then, a proximity network was not necessary. As an example, a modern computer numerical control machine may support direct numerical control for data input and use MTConnect for reporting results [17].

The access network serves as a means to transmit the collected data from the data collection sub-entity to the digital twin entity, and to transmit commands from the user entity to the device control sub-entity within the device communication entity. The most commonly used protocol for transmitting data in this network is Open Platform Communication Unified Architecture (OPC

Network	**Technology**		**Primary study**
Proximity	Protocol	Profinet	[P4], [P17]
		Modbus	[P8], [P11], [P13], [P16], [P17], [P19]
		MQTT	[P10]
		LoRaWAN	[P21]
Access	Protocol	OPC UA	[P2], [P3], [P4], [P7], [P8], [P12], [P14], [P15], [P16], [P17], [P18], [P20], [P21]
		MTConnect	[P4]
		MQTT	[P4], [P9], [P11], [P16]
		AMQP1.0	[P6]
		WebSocket	[P10]
		NC-link	[P14]
	Standard	IEC 61499	[P5]
		IEEE 1451	[P5]
	Architectural pattern	IDS	[P2]
	Open-source platform	Eclipse Hono	[P6]
Service	Protocol	OPC UA	[P3], [P18]
		MQTT	[P18]
	Open-source platform	Apache StreamPipes	[P21]
		Solace	[P10]
User	Protocol	OPC UA	[P3]
		HTTP	[P4], [P6], [P7], [P8], [P9], [P11], [P13], [P15], [P16], [P18], [P19], [P21]
		SMTP	[P10]
		WebSocket	[P6], [P17], [P20]
	Architectural pattern	IDS	[P9], [P15]
		REST	[P4], [P6], [P7], [P8], [P9], [P11], [P13], [P15], [P16], [P18], [P19], [P21]

Table 2. Technologies used for data transmission.

UA), which was utilised in more than 60% of the primary studies. Kim et al. motivate the adoption of OPC UA over other protocols due to its ability to facilitate integration across different platforms, timely detection of anomalies, and data security through user authorisation and authentication [P3]. Other primary studies aim to support publish/subscribe method for data exchange to enhance scalability utilised Message Queue Telemetry Transport (MQTT) [P4]. MQTT is a messaging protocol for IoT that defines a publish/subscribe messaging method [17]. Moreno et al. have advocated for the adoption of the International Data Space (IDS) architectural pattern, which is responsible for ensuring a secure and reliable channel of communication [P2]. In a similar vein, Rocha et al. have utilised standardised approaches such as IEEE 1451 and IEC 61499 to build an interoperable digital twin for monitoring water levels [P5]. The IEEE 1451 family of standards manage sensors and actuators of industrial systems, providing communication protocols for data acquisition and exchange that meet Industry 4.0 requirements. When combined with the IEC 61499 standard for data control and visualisation, it becomes a powerful tool for enhancing interoperability. Kherbache et al. propose the use of open-source platforms like Eclipse Hono for implementing the access network, which can eliminate protocol silos in the different OMEs [P6]. Eclipse Hono uses micro-services, called protocol adaptors, that map the supported protocols (e.g., HTTP, MQTT, or CoaP) to its API. This approach enables seamless integration and communication across multiple devices and protocols, facilitating interoperability and scalability in digital twin systems [P6]. When it comes to data representation in the access network, only a few primary studies (4/21) make use of an additional information model or data format on top of the protocol or standards used for data transmission. The most commonly used information model is AutomationML (AML). For instance, Fan et al. use AML to model all the components of a flexible manufacturing system [P14].

The service network is responsible for connecting sub-entities that offer different services within the digital twin core entity, such as operation and management, application and service sub-entity, and resource access and interchange. However, some current implementations of DT systems are designed as single private systems, in which services can communicate directly within the system without the need for a separate service network. Consequently, only a few papers (6/21) have implemented the service network in their digital twin systems. In such cases, data transmission protocols such as OPC UA and MQTT are used. Other primary studies have utilized open-source platforms to manage data transmission for different services within the digital twin, such as Apache StreamPipes [P21] and Solace [P10]. For example, Jacoby et al. utilized Apache StreamPipes to implement and manage multiple DT models. The primary motivation behind using Apache StreamPipes was its support for commonly used protocols, as well as the abundance of readily available implementations that can easily be customized with specific deep learning models, statistical analysis, or complex event processing [P21]. Open-source platforms are also preferred for data representation and management in the service network. Kherbache et al.

utilized Eclipse Ditto to model and manage data in the service network, motivated by its easy access to data [P6]. Similarly, Bamunuarachchi et al. utilized Eclipse rdf4j to support ontology models and RDF data [P13].

The user network connects the DT entity with third-party systems such as Enterprise Resource Planning (ERP) or Manufacturing Execution System (MES), allowing them to use the services provided by the DT [17]. Data transmission in the user network typically relies on protocols such as HTTP, WebSocket, SMTP, and OPC-UA. HTTP is the most commonly used protocol, cited in 57% (12/21) of primary studies. The WebSocket protocol is instead recommended in [P17] to support bidirectional communication for real-time data exchange. Some papers also suggest using the IDS architectural pattern in the user network to enable secure data exchange among different organisations [P9,P15]. The primary studies place significant emphasis on the technologies used for data representation and management within the user network. Information models, standards, graphical APIs, and query languages are some of the approaches identified. Asset Administration Shell (AAS) is favoured by around 30% (7/21) of the primary studies. AAS is employed to represent information related to physical assets and share it as a common information model with other stakeholders [P1]. Standards, e.g., ISO 10303 (STEP) and ISO 23952 (QIF) are used to represent CAD/CAM information [P4]. Assad et al. utilised WebGL, a JavaScript API, to render 2D/3D graphics [P17].

Network	**Technology**		**Primary study**
Access	Information model	AAS with eCl@ss dictionary	[P2]
		AutomationML	[P7], [P11], [P14]
	Data format	JSON-LD	[P5]
Service	Open-source platform	Eclipse Ditto	[P6]
		Eclipse rdf4j	[P13]
	Data format	PMML	[P3]
		JSON	[P10]
User	Information model	AAS	[P1],[P7],[P8], [P9],[P15],[P18],[P21]
		AutomationML	[P1]
		DTDL	[P1]
	Standard	ISO 10303 (STEP)	[P4]
		ISO 23952 (QIF)	[P4]
	Data format	JSON	[P6], [P11], [P18]
	Graphic API	WebGL	[P4], [P17]
		OpenGL	[P4], [P11]
	Query language	JSONata	[P8]
		SPARQL	[P13]

Table 3. Technology used for data representation and management.

4.2 Interoperability level (RQ2)

To answer this RQ, we investigated the LCIM levels of interoperability achieved for each network identified in the ISO 23247 standard. We used the LCIM descriptive view and our analysis involved examining the technologies utilised for data exchange and the requirements for achieving a particular level [32]. Table 4 presents a comprehensive summary of the LCIM level accomplished for each network, as well as the specific technology employed to achieve it. It is worth remarking that the LCIM levels are hierarchical, with each level encompassing all the capabilities of the lower levels. Consequently, we have reported in the table the highest level of interoperability was achieved and noted the lower levels

Network	LCIM level	Technology
Proximity	Syntactic	Profinet [P4], [P17], Modbus [P8], [P11], [P13], [P16], [P17], [P19], MQTT [P10], LoRaWAN [P21]
Access	Syntactic	MTConnect [P4], MQTT [P9], [P16], AMQP 1.0 [P6], Web-socket [P10], Eclipse Hono [P6]
	Semantic	OPC UA [P2], [P3], [P4], [P8], [P12], [P15], [P16], [P17], [P18], [P20], [P21], OPC UA + AutomationML [P7],[P14], MQTT + AutomationML [P11], AAS with dictionaries eCl@ss [P2], IEC 61499 and IEEE 1451 [P4], JSON-LD [P5]
Service	Semantic	OPC UA [P3], [P18], ApacheStreamPipes [P21], Eclipse Ditto [P6], Eclipse rdf4j [P13], PMML [P3], Solace [P10], JSON [P10]
User	Syntactic	HTTP [P6], [P11], [P16], [P19], SMTP [P10], WebSocket [P6], [P17], [P20]
	Semantic	AAS [P1], HTTP + AAS [P7], [P8], [P9], [P15], [P18], [P21], AutomationML [P1], DTDL [P1], ISO 10303 [P4], ISO 23952 [P4], HTTP + SPARQL [P13]

Table 4. Levels of interoperability reached

for the same network only if different technologies were employed. In addition, the same primary study might be encountered in different levels of interoperability for the same network if there were different options suggested to implement certain networks.

The proximity network uses a communication protocol, such as Profinet or Modbus for data transmission. These protocols define a specific syntax and format for the exchanged data, thereby fulfilling the requirements of syntax interoperability level. However, there is no evidence of using data models to define the meaning of the exchanged data in the proximity network, which would be the requirement to reach semantic interoperability level. Therefore, the highest interoperability level reached in the proximity network is syntax interoperability.

In the access network, 28% (6/21) of the primary studies reached syntactic interoperability using a communication protocol (such as MTConnect, and AMPQ 1.0). However, the majority of the implementations for the access network 66% (14/21) achieved semantic interoperability, where systems exchange data that can be semantically parsed. This was accomplished by employing OPC UA and information models such as AAS and AutomationML or standards such as IEC 61499 and IEEE 1451. In the manufacturing and automation domain, OPC UA is emerging as a universal standard protocol for achieving semantic interoperability among connected systems [P12]. OPC UA offers extendable information models for a range of application domains, enabling semantic interpretation of encoded information. It goes beyond being just a transport protocol for industrial applications and provides a comprehensive set of services and functionalities to support secure and reliable communication between different components of a distributed system [14]. However, to achieve full semantic interoperability, the information models should be well-defined and consistent. Moreno et al. used AAS in combination with eCl@ss dictionary to standardise information models and achieve semantic interoperability in the access network [P2]. The Reference Architecture Model for Industry 4.0 presents the concept of AAS as the foundation for interoperability, which is defined as a digital representation of an asset [P7]. The use of standardised dictionaries such as eCl@ss can simplify the task of assigning semantic descriptions to the information models [P2].

In the service network, semantic interoperability is achieved through all the reported approaches by the use of open-source platforms and the OPC UA protocol. Kim et al. proposed a DT architecture based on the ISO 23247 RA for anomaly detection, with the main services that communicate through the service network being the data presentation and anomaly detection and prediction services [P3]. Communication is based on OPC UA, which gathers data in information models. The pre-processed OPC UA data is then used to generate convolutional neural network (CNN)-based real-time anomaly detection and prediction DTs. Finally, the model is converted to Predictive Model Markup Language (PMML). All technologies used in the service network support semantic interoperability [P3]. Jakoby et al. use the ApacheStreamPipes platform to manage services, which also supports a semantic description level [P21]. Other open-source platforms that support semantic interoperability used in the service network are Eclipse Ditto [P6] and Eclipse rdf4j [P13]. Eclipse Ditto and Eclipse rdf4j both utilize semantic technologies such as JSON-LD, RDF, and SPARQL to enable devices and systems to achieve semantic interoperability. By using a shared data representation, they facilitate communication and integration regardless of underlying technologies or protocols.

Our analysis showed that 43% (7/16) of the studies addressing the user network used a protocol for data transmission such as HTTP [P16,P19], SMTP [P10]. Alternatively, they used WebSocket [P17,P20] and JSON [P6,P11] as a data format. In these cases, syntactic interoperability is reached. Conversely, other primary studies (57% or 9/16) reached semantic interoperability by leveraging various information models, standards, and semantic technologies. The most used information model is AAS, commonly implemented over HTTP APIs. However, we found no evidence of methods or taxonomies employed to enable interoperating systems to anticipate the context of exchanged data, a prerequisite for achieving a higher level of interoperability such as pragmatic interoperability.

5 Summary, discussion and future directions

In this section, we summarise and discuss our findings on the technologies used for data exchange and the level of interoperability achieved. Table 5 gives an overview of our findings and serves as a prescriptive tool for each network. In particular, the table points out the levels of interoperability reached along with the most commonly used technology. The technologies corresponding to each level of interoperability, except for the technical level, are documented in the primary studies. For the technical level, specifications from the ISO 23247 standard are included since this information was not available in the primary studies. The interested reader can check the detailed findings in the replication package [3]. In addition, Table 5 highlights levels of interoperability that may be desirable to achieve along with possible technologies for achieving them (marked with the blue italic text). We have determined these based on motivating examples found in the primary studies as well as our consolidated experience in collaborative research projects on DTs. For the empty cells of Table 5, we did not find evidence regarding the technologies employed to attain specific levels of interoperability or motivation for their necessity. For the cells highlighted in grey, we reason that

there is no need for reaching higher interoperability levels. Using these findings and LCIM requirements as a basis, we discuss the needs and trade-offs involved in attaining higher levels of interoperability.

Our analysis has shown that the highest level of interoperability reached in the proximity network is syntactic interoperability. This is in line with the primary purpose of this network, which is to collect data from physical entities and transmit it to the data collection entity, without significant processing of the data occurring at this stage. The Modbus protocol is commonly employed in the proximity network to achieve syntactic interoperability, typically over industrial Ethernet or proprietary networks. Although it is possible to achieve semantic interoperability in the proximity network, we believe that it may not be necessary or desirable given the network's primary purposes. Achieving higher levels of interoperability would require some form of reasoning and language, which could potentially impact the timeliness of communication. In this case, achieving higher levels of interoperability at the expense of performance and efficiency may not be worthwhile, even if possible (grey cells in Table 5). Our analysis has

	Technical	**Syntactic**	**Semantic**	**Pragmatic**	**Dynamic**	**Conceptual**
Proximity	Industrial Ethernet or proprietary network	Modbus				
Access	LAN/ WLAN or cellular network	MQTT	OPC UA			
Service	Wired IP-based protocols	OPC UA	OPC UA	*SOA & microservices*		
User	Internet or private intranet	HTTP	AAS	*Linked data and ontologies*	*Linked data and ontologies*	*Linked data and ontologies*

Table 5. Various LCIM levels achieved per ISO 23247 network along with the most used technology.

shown that the highest level of interoperability achieved for all other networks (including access, service, and user) is semantic interoperability. While OPC UA is used to achieve semantic interoperability in the access and service networks, AAS is the most common technology used in the user network. Although our analysis did not identify any studies that explicitly discussed the need for achieving higher interoperability levels in the access and service networks, we believe that pragmatic interoperability is needed in the service network. Pragmatic interoperability requires that the systems can exchange information describing the services along with their availability. The service network provides means of communication for sharing DT applications and services including simulation, analysis of data captured from OMEs, and reporting production status [17]. Hence, pragmatic interoperability in the service network seems to be not only desirable but needed for ensuring the functional suitability of DTs. Several studies have explored the use of Service-Oriented Architecture (SOA) and micro-service patterns as promising solutions for achieving pragmatic interoperability [12]. At these networks, achieving dynamic or conceptual interoperability may nega-

tively impact other DTs' qualities namely security and privacy as we discuss in the following paragraph [3].

In the user network, our analysis has revealed the need for higher levels of interoperability than semantic in situations where data is exposed to external systems or other DTs. A use case provided by Kuruppuarachchi et al. in the additive manufacturing domain highlights this need [P1]. In this scenario, a product owner has contracted several manufacturers to produce parts for their product, and each manufacturer has their own DT for their product part. The manufacturers share product-related information with a collaborative DT system. The goal is to optimise the production line, avoid downtime, and reduce costs by modifying individual manufacturing capabilities based on the states of other manufacturers. However, achieving this requires at least dynamic interoperability. One example of achieving dynamic interoperability using linked data and ontologies in the domain of System of System has been proposed by Axelsson [3], which could be applicable in this case. To achieve higher levels of interoperability beyond semantic, it is necessary for DTs exchanging data to have access to each other information regarding properties and functions, and to interpret data in light of this information. While this can be a challenging task for many software-intensive systems, it is even more daunting for DTs due to several reasons. To begin with, it can be difficult to determine which data from DT subsystems or other DTs should be shared and for how long it needs to be stored. As a result, this can lead to gathering an excessive amount of data, ultimately resulting in decreased performance [15]. In situations when interoperating DTs are owned by different organisations, granting access to other DTs' internal data may be difficult due to confidentiality, accuracy and trust, and security [3]. This is particularly true when internal data holds significant value (e.g., commercial competitive situations) and its manufacturer may be hesitant to share it with external organisations [3]. Moreover, even if the data is shared, it may be difficult for an external organisation to verify its accuracy and reliability [3]. Ensuring safe and secure storage of such data is also essential [3]. Eventually, there may be situations in which data can not be shared due to privacy regulations.

Web Ontology Language (OWL) and Resource Description Framework (RDF), can help mitigate some of the above-mentioned challenges associated with high levels of interoperability [16]. Hence, further research is needed to develop and refine existing standards to meet the evolving needs of complex and diverse DTs taking into account OWL and RDF for enhancing interoperability.

6 Related work

Numerous studies have explored ways to improve interoperability both within and between DT systems. Nonetheless, to the best of our knowledge, current research appears to lack a comprehensive analysis and assessment of the interoperability requirements and support for DTs. Li et al. presented a framework for achieving seamless interaction between the physical and virtual spaces of a single DT system [23]. To uniformly model all manufacturing units, they proposed a semantic modelling methodology. Damjanovic-Behrendt and Behrendt have stressed the importance of utilizing open-source technologies to facilitate

interoperability in DTs [11]. To this end, they have introduced a collection of the most significant open-source tools and technologies for designing and building DTs in the context of smart manufacturing [11]. In addition, the authors have discussed the use of a micro-service architecture for the DT demonstrator to support semantic data interoperability. Marie Platenius-Mohr et al. investigated the interoperability issue of DTs among various organisations [27]. Their main objective was to identify a solution for achieving interoperability between different DTs. Park et al. made a significant contribution to enhancing the interoperability of DTs with external systems in smart manufacturing [25]. They proposed a new data schema to incorporate existing standards for smart manufacturing, ensuring interoperability. The authors also developed a cloud-based DT that uses the proposed schema, which is interoperable with existing legacy systems. Ariansynah and Pardamean have highlighted the importance of integrating asset prognostic and health monitoring DTs with other software systems that manage business operations such as CMMS and ERP to minimise asset downtime in a cost-effective and time-efficient manner [2]. To address this issue, they proposed using rule-based ontology modelling and reasoning to improve the interoperability of DTs with CMMS and ERP systems. Cavalieri and Gambadoro presented a novel approach for enhancing the semantic interoperability of digital twins (DTs) with external systems [7]. The primary objective of their research was to establish communication between DTs based on Digital Twins Definition Language (DTDL) and any applications that conform to the Open Platform Communications Unified Architecture (OPC UA).

7 Conclusion and future work

This paper investigates interoperability in digital twin software architectures for manufacturing. We analysed 21 primary studies selected from an initial pool of 2403 peer-reviewed publications. Through an examination of the data extracted from these primary studies, we identified the specific technologies employed for data exchange in DTs, as well as the degree of interoperability that was achieved during such exchanges. Our analysis has revealed that current DT architectures are successful in achieving semantic interoperability. However pragmatic and dynamic interoperability levels are desirable, particularly in federated DTs. Achieving higher levels of interoperability in federated DTs presents challenges related to accuracy, trust, security, and privacy. To overcome these challenges, standards and standardised semantic mapping frameworks hold promise.

Concerning future works, one area of interest involves exploring the potential of linked data and ontologies for achieving pragmatic, dynamic and conceptual interoperability. Also, we aim to focus on defining a standards-based architectural framework for digital twins in manufacturing that facilitates data interoperability. This framework will consist of several views, including the functional view, technical adaptation and implementation view, and interoperability view.

Acknowledgements

The work in this paper has been supported by the Swedish Knowledge Foundation (KKS) through the ACICS and Modev projects, by the Excellence in Production Re-

search (XPRES) Framework, by the EU - NextGenerationEU under the Italian MUR National Innovation Ecosystem grants ECS00000041 – VITALITY, and PE0000020 – CHANGES. The authors also acknowledge the support of the MUR (Italy) Department of Excellence 2023 - 2027 for GSSI.

Primary Studies

P1. P. M. Kuruppuarachchi et al. "Trusted and secure composite digital twin architecture for collaborative ecosystems," *IET Collab. Intell. Manufacturing*, v 5, n. 1, 2023.
P2. T. Moreno et al. "Scalable digital twins for industry 4.0 digital services: a dataspaces approach," *Production & Manufacturing Research*, vol. 11, no. 1, p. 2173680, 2023.
P3. D. B. Kim et al. "A digital twin implementation architecture for wire+ arc additive manufacturing based on iso 23247," *Manufacturing Letters*, vol. 34, pp. 1–5, 2022.
P4. A. Farhadi et al. "The development of a digital twin framework for an industrial robotic drilling process," *Sensors*, v.22, n.9, 2022.
P5. H. da Rocha et al. "An interoperable digital twin with the ieee 1451 standards," *Sensors*, vol. 22, no. 19, p. 7590, 2022.
P6. M. Kherbache et al. "Digital twin network for the iiot using eclipse ditto and hono," *IFAC-PapersOnLine*, vol. 55, no. 8, pp. 37–42, 2022.
P7. K. Ding et al. "Aml-based web-twin visualization integration framework for dt-enabled and iiot-driven manufacturing system under i4. 0 workshop," *Journal of Manufacturing Systems*, vol. 64, pp. 479–496, 2022.
P8. F. Schnicke et al. "Architecture blueprints to enable scalable vertical integration of assets with digital twins," in *IEEE 27th Int. Conference ETFA*. IEEE, 2022, pp. 1–8.
P9. F. Yallıç et al. "Asset administration shell generation and usage for digital twins: A case study for non-destructive testing by x-ray."
P10. V. Leiras et al. "Iso23247 digital twin approach for industrial grade radio frequency testing station," in *IEEE 27th Int. Conference ETFA*. IEEE, 2022, pp. 1–8.
P11. G. N. Schroeder et al. "A methodology for digital twin modeling and deployment for industry 4.0," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 556–567, 2020.
P12. A. Redelinghuys et al. "A six-layer architecture for the digital twin: a manufacturing case study implementation," *Journal of Intelligent Manufacturing*, vol. 31, pp. 1383–1402, 2020.
P13. D. Bamunuarachchi et al. "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.
P14. Y. Fan et al. "A digital-twin visualized architecture for flexible manufacturing system," *Journal of Manufacturing Systems*, vol. 60, pp. 176–201, 2021.
P15. M. Jacoby et al., "An approach for industrie 4.0-compliant and data-sovereign digital twins," *at-Automatisierungstechnik*, vol. 69, no. 12, pp. 1051–1061, 2021.
P16. F. Pires et al. "Decision support based on digital twin simulation: a case study," in *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2020*. Springer, 2021, pp. 99–110.
P17. F. Assad et al. "Utilising web-based digital twin to promote assembly line sustainability," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2021, pp. 381–386.
P18. M. Redeker et al. "A digital twin platform for industrie 4.0," in *Data Spaces: Design, Deployment and Future Directions*. Springer International Publishing Cham, 2022, pp. 173–200.
P19. E. Russo et al."Lidite: a full-fledged and featherweight digital twin framework," *IEEE Trans. on Dependable and Secure Computing*, 2023.
P20. K. Židek et al. "Digital twin of experimental smart manufacturing assembly system for industry 4.0 concept," *Sustainability*, v.12, n.9, 2020.
P21. M. Jacoby et al. "An approach for realizing hybrid digital twins using asset administration shells and apache streampipes," *Information*, vol. 12, no. 6, p. 217, 2021.

References

1. Ali, N.B., Petersen, K.: Evaluating strategies for study selection in systematic literature studies. In: Procs of ESEM (2014)
2. Ariansyah, D., Pardamean, B.: Enhancing interoperability of digital twin in the maintenance phase of lifecycle. In: Procs of ICITISEE. IEEE (2022)
3. Axelsson, J.: Achieving system-of-systems interoperability levels using linked data and ontologies. In: INCOSE International Symposium. vol. 30, pp. 651–665. Wiley Online Library (2020)
4. Basili, V.R., Caldiera, G., Rombach, H.D.: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering (1994)

5. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software (2007)
6. Carney, D., Oberndorf, P.: Integration and interoperability models for systems of systems. Tech. rep., Carneige-Mellon Univ Pittsburgh PA Software engineering Inst (2004)
7. Cavalieri, S., Gambadoro, S.: Proposal of mapping digital twins definition language to open platform communications unified architecture. Sensors **23**(4), 2349 (2023)
8. Charmaz, K., Belgrave, L.L.: Grounded theory. The Blackwell encyclopedia of sociology (2007)
9. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: Procs of ESEM (2011)
10. Dalibor, M., Jansen, N., Rumpe, B., Schmalzing, D., Wachtmeister, L., Wimmer, M., Wortmann, A.: A cross-domain systematic mapping study on software engineering for digital twins. Journal of Systems and Software p. 111361 (2022)
11. Damjanovic-Behrendt, V., Behrendt, W.: An open source approach to the design and implementation of digital twins for smart manufacturing. International Journal of Computer Integrated Manufacturing **32**(4-5), 366–384 (2019)
12. Ferko, E., Bucaioni, A., Behnam, M.: Architecting digital twins. IEEE Access **10**, 50335–50350 (2022)
13. Ferko, E., Bucaioni, A., Pelliccione, P., Behnam, M.: Standardisation in digital twin architectures in manufacturing. In: 2023 IEEE 20th International Conference on Software Architecture (ICSA). pp. 70–81 (2023)
14. Graube, M., Hensel, S., Iatrou, C., Urbas, L.: Information models in opc ua and their advantages and disadvantages. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–8. IEEE (2017)
15. Henningsson, K., Wohlin, C.: Understanding the relations between software quality attributes-a survey approach. In: Proceedings 12th International Conference for Software Quality (2002)
16. Herzog, R., Jacoby, M., Podnar Žarko, I.: Semantic interoperability in iot-based automation infrastructures: How reference architectures address semantic interoperability. at-Automatisierungstechnik **64**(9), 742–749 (2016)
17. International Organization for Standardization: Iso 23247-1:2021, `https://www.iso.org/standard/75066.html`
18. ISO: ISO/IEC 25010, `https://iso25000.com/index.php/en/iso-25000-standards/iso-25010`
19. Kitchenham, B., Brereton, P.: A systematic review of systematic review process research in software engineering. Information and software technology (2013)
20. Kuruppuarachchi, P., Rea, S., McGibney, A.: An architecture for composite digital twin enabling collaborative digital ecosystems. In: Proc of CSCWD. pp. 980–985. IEEE (2022)
21. Lattanzi, L., Raffaeli, R., Peruzzini, M., Pellicciari, M.: Digital twin for smart manufacturing: A review of concepts towards a practical industrial implementation. International Journal of Computer Integrated Manufacturing **34**(6), 567–597 (2021)
22. Leal, G.d.S.S., Guédria, W., Panetto, H.: Interoperability assessment: A systematic literature review. Computers in Industry **106**, 111–132 (2019)
23. Li, J., Zhang, Y., Qian, C.: The enhanced resource modeling and real-time transmission technologies for digital twin based on qos considerations. Robotics and Computer-Integrated Manufacturing **75**, 102284 (2022)
24. Lidell, A., Ericson, S., Ng, A.H.: The current and future challenges for virtual commissioning and digital twins of production lines. In: Proc of SPS2022 (2022)
25. Park, Y., Woo, J., Choi, S.: A cloud-based digital twin manufacturing system based on an interoperable data schema for smart manufacturing. International Journal of Computer Integrated Manufacturing **33**(12), 1259–1276 (2020)
26. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Procs of EASE (2008)
27. Platenius-Mohr, M., Malakuti, S., Grüner, S., Schmitt, J., Goldschmidt, T.: File- and api-based interoperability of digital twins by model transformation: An iiot case study using asset administration shell. Future Generation Computer Systems **113**, 94–105 (2020)
28. Qi, Q., Tao, F.: Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. Ieee Access **6**, 3585–3593 (2018)
29. Shull, F., Singer, J., Sjøberg, D.I.: Guide to advanced empirical software engineering. Springer (2007)
30. Stol, K.J., Ralph, P., Fitzgerald, B.: Grounded theory in software engineering research: a critical review and guidelines. In: Proc. of the 38th Int. conference on software engineering (2016)
31. Tolk, A.: The elusiveness of simulation interoperability – what is different from other interoperability domains? In: 2018 Winter Simulation Conference (WSC). pp. 679–690 (2018)
32. Wang, W., Tolk, A., Wang, W.: The levels of conceptual interoperability model: applying systems engineering principles to m&s. arXiv preprint arXiv:0908.0191 (2009)
33. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer (2012)
34. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Procs of EASE (2014)
35. Wohlin, C., Höst, M., Henningsson, K.: Empirical research methods in software engineering. In: Empirical methods and studies in software engineering, pp. 7–23. Springer (2003)