Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 758–763

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Towards a Requirement-driven Digital Twin Architecture

Chukwudi Nwogu[a*], Giovanni Lugaresi[b], Anastasia Anagnostou[a], Andrea Matta[b], Simon J.E. Taylor[a]

[a]*Brunel University London, Uxbridge, UB8 3PH, UK*
[b]*Politecnico di Milano , Via La Masa 1, 20156 Milano , ITALY*

* Corresponding author. Tel.: +44-189-527-4000 ; fax: +44-189-523-2806. *E-mail address:* chukwudi.nwogu@brunel.ac.uk

Abstract

The fourth industrial revolution introduces new technological innovations and amplifies the importance of some existing technologies such as Artificial Intelligence (AI), Internet of Things (IoT), and Blockchain, amongst others. Digital Twin (DT) has emerged as one of the most prominent technologies of this era and has caught the attention of the industry, academia, and governments. However, the realisation of the full potential of DT is challenged by the lack of standard terminologies and practices, amongst others. This is reflected in the state-of-the-art in DT architectures, which indicates that there is no widely accepted DT framework. Literature on DT architecture is dominated by application- and/or technology-specific architectures with components and connectors that not only vary extensively but are named differently. The use of different terminologies for components could hinder the ability to identify commonality in frameworks and makes it difficult for new entrants in the field to find guidance. Also, literature does not clarify on the connection between the requirements and the components of a DT architecture. To address these problems, this paper proposes a requirement-driven, technology-agnostic DT architecture that consists of standard components that can be traceable to the definitions, requirements, and mandatory functionalities of DT captured in existing literature. The architecture can be applied to various fields and uses cases, based on their respective needs. The paper aims at providing guidance for developing digital twin architectures for a flexible spectrum of applications.

Keywords: Digital Twin; Industry 4.0; Cyber-physical systems; Internet-of-Things

1. Introduction

Although there are several definitions of Digital Twin (DT) in literature [9, 28, 12, 32], none has been officially endorsed. This study will adopt the definition of DT by van der Valk et al. [34], who described a DT as *"a virtual construct that represents a physical counterpart, integrates several data inputs with the aim of data handling, data storing, and data processing, and provides an automatic, bi-directional data linkage between a virtual world and a physical one. Synchronization is crucial to the Digital Twin to display any changes in the state of the physical object. Additionally, a Digital Twin must comply with data governance rules and must provide interoperability with other systems"*. This definition encapsulates most of the mandatory characteristics of a DT that have been noted in literature, such as synchronization, bi-directional communication, control, anomaly detection and diagnostic, optimization [12, 20, 32]. These attributes differentiate a DT from other digital forms, such as Digital Model and Digital Shadow; the difference can be described in terms of the data exchange between digital forms and their physical counterparts [12]. For example, a Digital Model exchanges data with its physical counterpart manually in both directions; while for a Digital shadow, the data exchange is automated from the physical to the virtual counterpart, but manual from the virtual to the physical counterpart. In contrast, a DT has an automated bidirectional data exchange between its virtual and physical entities.

Typically, a DT acquires data from its real-world entity, and manipulates the data to provide services, which have the potential of improving its real-world counterpart, as well as itself. In order to improve or effectively reflect the behaviour of its real-world entity, a DT should have the capability to automate data acquisition [6], and this should be a fundamental element of a DT architecture [33]. Given that a DT acquires data from multiple sources, including offline data sources (e.g. databases, stored data, external data, maintenance logs, etc.), there is an emerging case for semi-manual data acquisition [34].

2212-8271 © 2022 The Authors. Published by Elsevier B.V.

10.1016/j.procir.2022.05.058

However, there are no practical approaches for implementing data acquisition and this has been identified as (1) one of the major challenges of realising DT and (2) a source of cost increase in DT development (Dittmann et al. [6]). These authors suggest that this stems from the absence of guidance on the framework for developing DT architectures. This may hinder the rapid development and adoption of DT, given that there is no standardization or a uniform description of DT architectural components. Steindl et al. [31] report that the application of DTs to various domains and at different stages in the lifecycle of their real-world entities is testament to the lack of a clear definition of the capabilities and concepts of DT. The authors argue that this results in case study-driven, varying interpretations of DT and lack of templates for DT architectures. Therefore, agreeing on DT components and data analysis techniques is paramount to advancing DT technology [24].

To address the aforementioned gap, this paper will undertake the following steps. Firstly, it will review literature on existing DT architectures and match the architectures to the mandatory functionalities of DT identified in literature. Secondly, it will describe the requirements in terms of attributes that distinguish a DT from other digital forms. Finally, the study will make a connection between the requirements and components of a DT to propose a requirement-driven DT architecture. The architecture consists of standard components that can be traceable to the definitions, requirements, and 'mandatory' functionalities of a DT, derived from existing literature. In order to achieve the above-stated objectives, this paper will seek to answer the following questions: (1) What are the requirements and mandatory functionalities of a DT? (2) How can these requirements and functionalities form a basis for developing a domain- and technology-agnostic DT architecture?

The rest of the paper is structured as follows: Related literature is presented in Section 2. Section 3 presents the proposed architecture. Discussion on aligning DT requirements to DT architectural components is covered in Section 4. The conclusion and further developments follow in Section 5.

2. State of the Art

This section will cover various contributions in literature on DT architecture, and that will be followed by a brief discussion on the research gap identified.

2.1. Existing Architectures

Lee et al. [13] develop 5C, a cyber physical system architecture for Industry 4.0-based manufacturing system, which provides plug and play smart connection, offers smart analytics for subsystem health, enables DT models for components and machines, supports decision-making using cognition, and achieves resilience using self-configuration.

Alam and El Saddik [1] work on a DT architecture called C2PS (cloud-based cyber-physical systems) extends 5C architecture by employing cloud technology in the cyber, cognition and configuration levels. The key contribution of C2PS is that every physical entity is associated with a cloud-hosted cyber entity, such that two entities can establish peer-to-peer (P2P) connections via direct physical connection or via indirect cloud-based DT connections.

A five-layered architecture is proposed by Josifovska et al. [11] as part of the framework for developing DTs of CPS. The framework consists of the Physical Entity, Virtual Entity, Data Management, and Service Platforms.

Borangiu et al. [3] present a four-layered architecture in which every layer on top of the physical system is a DT Layer. The four layers of the architecture are the data acquisition and transmission twins, virtual twins of subprocesses, predictive twins, and decision-making twins.

Souza et al. [30] propose an Industrial Internet of Things-(IIoT)-based DT architecture, which consists of Internal Server Layer and IIoT Gateway Layers. The internal server is the computer system that runs the DT and simulations, while the IIoT Gateway is the channel for communication between the DT and its physical twin.

Ghita et al. [8] develop a three-layered DT architecture, which consists of industrial, application, and communication layers. The industrial layer represents the physical system, the application layer focuses on digital components of the architecture and their features, and the communication layer is concerned with the interaction between the DT and the physical system.

Nwogu et al. [22] present a symbiotic simulation system-based architecture for digital twin. The DT Layer of the architecture consists of data acquisition, analytics, scenario manager, optimisation and symbiotic simulation modules (SSM). The use of SSM and MQTT make the architecture technology-specific.

Redelinghuys et al. [25] study a six-layer architecture for DT that allows for hardware from various vendors to be used in both the physical and digital worlds. According to the authors, the architecture was inspired by Lee et al. [13] and consists of a physical twin in Layers 1 and 2. Layer 3 is a vendor-neutral communication medium, which acts as the local data repository for collecting the sensor data from a controller in Layer 2. IoT gateway or data-to-information converter is in Layer 4; while Layers 5 and 6 consist of a cloud repository, and emulation and simulation tools, respectively.

In one of the most recent DT architectures, Vrabic et al. [36] offer an intelligent agent-based architecture for resilient digital twin in manufacturing. The architecture employs engineering resilience principles to improve the digital twin's ability to represent its real-world entity.

Farsi et al. [7] describe a DT architecture for product lifecycle cost (LCC) estimation, which synchronizes between the physical and digital worlds using an ontology-based approach. The DT architecture aims at supporting the reduction in product cost and improving product efficiency.

In order to optimize the productivity of Controlled Environment Agriculture (CEA), Chaux et al. [4] propose a three-layer DT architecture, which consists of the physical asset, the digital twin, and an intelligent layer. The DT architecture was aimed at building a DT that has the capability for bi-directional communication, in which a simulation model is used to optimize pro-

Table 1. Features of available DT architectures from the literature.

Reference	Sync.	Bi-Dir. Info	Diagnostic	Control	Predictive/Prescr.	Monitoring	Optimization
Alam and El Saddik [1]	•	•		•			
Borangiu et al. [3]	•	•	•		•	•	•
Chaux et al. [4]	•	•					•
Farsi et al. [7]	•						•
Ghita et al. [8]		•				•	
Josifovska et al. [11]			•		•	•	•
Lee et al. [13]	•		•		•	•	•
Lui et al. [16]			•		•	•	•
Mourtzi et al. [19]						•	•
Nwogu at al. [22]	•	•	•	•	•	•	•
Redelinghuys et al. [25]	•		•		•	•	
Souza et al. [30]	•		•	•		•	
Steindl et al. [31]		•	•	•	•	•	
Vrabic et al. [36]	•		•		•	•	•

ductivity. The plan for future work on the architecture includes optimising crop treatment and climate control strategy using the simulation models from the DT.

Mourtzis et al. [19] develop a DT architecture for Fused Decomposition Modelling (FDM), which uses common process data to support engineers in performing online and offline simulation. The idea of the framework is to integrate a DT and Augmented Reality (AR) for quality improvement of 3D parts and reduction in human-related error resulting from incorrect machine setup.

2.2. Research Gap

Table 1 matches the reviewed DT architectures to the core functionalities of DT identified in literature. The table shows that majority of the architectures did not completely align with the functionalities of DT, probably since most contributions are domain- and use case-specific. The vast majority of the architectures discussed above is not only case-driven, but varies extensively in terms of their respective components. Going by the definition of DT adopted by this study, it can be argued that some of the DT architectures may not meet the requirements of fully fledged DTs.

3. Digital Twin Requirements

According to several studies [17, 12, 32, 20], there is a set of core attributes that differentiate a DT from other digital forms. These include, in no particular order, synchronization, learning and adaptability, bidirectional information flow, monitoring capability, predictive and prescriptive capabilities, and optimization. These will be described in the following section.

3.1. Synchronization

The synchronisation attribute allows a DT to be dynamic enough to reflect the state of its real-world twin at all times. If such an alignment is guaranteed, any evaluation that is carried out in real-time can be considered reliable; for instance, the comparison of two production policies. The challenge with the ability of a DT to achieve synchronisation is that the state of a physical entity changes quickly over time, and optimisation activities could be referred to system states that differ from the current one [23]. To avert this, a continuously synchronised parent model can be maintained, from which many child models can be generated and run when required [10].

3.2. Learning and Adaptation

Most physical systems are dynamic and change with a high frequency (e.g., flexible manufacturing systems); hence, the digital constructs need to be able to adapt to always represent the physical system. Types of adaptation may include: (1) adaptation of the model structure, which refers to the logical layout and material flows; those can be either generated or adapted from a previously available model. (2) the tuning of the model level of detail refers to the possibility to exclude from the digital representation the components that do not significantly contribute to estimating the performance of the system, with respect to a particular goal [18]. (3) the adjustment of model parameters to reflect the current conditions.

3.3. Bi-directional Information Flow

In accordance with the definition by Kritzinger et al. [12], a DT can be called as such if information flow is not only from a real-world entity to a digital system, but also vice versa. Accordingly, decisions that are taken within a digital system (e.g., optimization of a production schedule), will be applied to the corresponding real-world entity, automatically through a control system (e.g., an actuator). Bidirectional communication from a DT to a Physical Twin may not always be used to control the Physical Twin, rather it could represent additional inspection or data collection activities [35].

3.4. Monitoring Capabilities

Thanks to a real-time information flow from a physical system, a DT can monitor the condition of its physical twin in real-time. For instance, a machine tool vibration data can be elaborated and used to produce the health score of a resource in use, in real-time. Damjanovic-Behrendt and Behrendt [5] propose monitoring services that divide the system into micro-services or smaller applications, which enable a DT to monitor the various aspects of its Physical Twin, such as tracing, performance metrics, alerting (i.e. providing the capability to detect and isolate problems) and dashboard services.

3.5. Predictive and Prescriptive Capabilities

A DT may include digital models that enable forward-looking analyses. For instance, the real-time state of a manufacturing system can feed a Discrete Event Simulation Model, used to estimate the end-of-the-day production performance starting from current conditions and the expected number of orders. By coupling the predictive capabilities with the bi-directional information flow, it may be stated that a DT has also prescriptive capabilities, since it can use the knowledge acquired in forward-looking scenarios to generate actionable commands in a real system.

3.6. Optimization

A DT-based optimisation system offers a rapid identification of optimal solutions [15]. The possibility of evaluating scenarios that are not yet applied in the real system means that a DT is enabled to search for an optimal configuration of real system settings. For instance, a scheduling plan may be investigated in search of a plan that minimizes the number of resources in use, and consecutively, the energy consumption, with the aim of increasing the sustainability score of the company. Among others, Leng et al. [15] propose a DT-driven joint optimisation solution to optimise the utilisation and efficiency of a warehouse product service system (PSS).

4. DT Requirement-driven Architecture

Based on the requirements and characteristics of a DT identified in literature and in Section 3, this study proposes a requirement-driven DT architecture, as shown in Figure 1. Table 2 shows the connection between DT requirements and the components of the proposed architecture, which has three major parts, namely: (1) the physical twin, (2) the communication or integration medium, and (3) the digital twin system.

4.1. Physical Twin

The physical twin (PT) is the real-world physical or perceived system, for instance, a manufacturing system or an airport process, which is dynamically connected with the DT via a communication or integration medium.

4.2. Communication or Integration Medium

The communication or integration medium of a DT is typically implemented using communication protocols with the capability of providing a bi-directional communication between the PT and DT. A bi-directional data flow distinguishes a DT from other digital forms such as a, digital model, digital shadow, etc. [12]. The requirements of a use case, including the type of data to be transferred, determine the type of devices used and the communication strategy and protocols that apply. Such protocols vary extensively and are not the focus of this paper.

4.3. Digital Twin System

The DT system consists of the Data Acquisition Module, Data Analytics Module, Database Module, Simulation Module, Scenario Manager, Optimisation Module, and Controls Module.

4.3.1. Data Acquisition Module

To improve the real-world entity it is representing or effectively reflect its behavior, a DT should have the capability to automate data acquisition [6]; hence, the Data Acquisition Module is one of the most crucial components of a DT [33]. The data exchange between a DT and its PT can be volatile (e.g., real-time sensors transmitting the parts location, activity timestamps in a manufacturing facility, movement of passenger in an airport, etc.) or static (e.g., list of manufacturing equipment, list of airlines or number/location of check-in desks) [5]. The data requirements of a DT drive the implementation of the Data Acquisition Module, which can be implemented as a web service, web application [26, 21] or mobile application [23].

4.3.2. Data Analytics Module

The Data Analytics Module is the engine of a DT, which applies various analytics techniques to provide the mechanism to manage, fuse and process the vastly heterogeneous data acquired by the DT System. Analytics techniques such as streaming process, time-series-based, batch-oriented, and security analytics processing, etc., offer feedback mechanisms for decision-making and control of the Physical Twin, as well as provide results that become input to simulation and visualisation [5]. Also, simulation outputs can be analysed using data analytics techniques.

4.3.3. Database Module (DB)

The Database Module encompasses the internal repository or any form of data storage mechanism that houses the data model of a DT. The DB can be cloud-based and this offers benefits such as accessibility, scalability, processing power and efficiency in data transfer [29, 1]. A local-based DB may be used for the security of sensitive data, but a hybrid approach consisting of a combination of local and offsite DB (e.g., cloud, Edge, Fog, Mainframe) may be employed in practice [35]. For instance, for use cases involving safety critical or personal sensitive data, the data can be pre-processed by an Edge Computing-

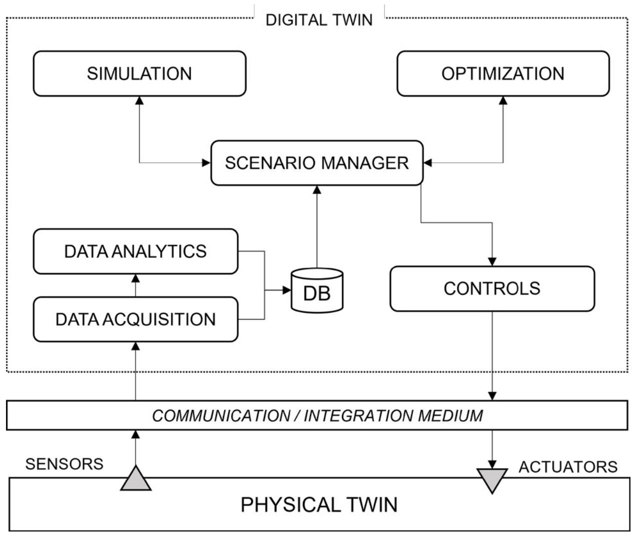

Fig. 1. Exemplary DT architecture.

Table 2. Connection between the DT requirements and the components of the proposed architecture.

Requirements	Components
Synchronization	Scenario Manager; Database Module
Learning	Simulation Module; Database Module
Bi-directional Information Flow	Sensors; Integration Module (Controllers); Actuators
Monitoring Capabilities	Data Acquisition Module; Data Analytics Module
Predictive/Prescriptive Capabilities	Data Acquisition Module; Data Analytics Module; Database Module; Simulation Module
Optimization	Optimization Engine; Scenario Manager; Simulation Module

based system to stripe out the sensitive data before loading it to the cloud system. Edge Computing-based system also provides resilience to a Cloud System performance issue or failure.

4.3.4. Scenario Manager

The Scenario Manager carries out what-if analyses based on the outcome of data analytics and/or simulation experimentation. It can control the running of simulation experimentation in order to meet the objective of the experimentation [23]. For instance, by monitoring the production performance over a certain amount of time, the scenario manager may decide to launch forward looking simulation models to evaluate the performance of different policies under the current set of parameters.

4.3.5. Simulation Module

Depending on the requirements, the simulation module can be implemented using either discrete event, agent-based, system dynamics (i.e., continuous simulation), or hybrid simulation [27]. As discussed in the Data Analytics Section, input to or output from simulation can support data analytics and/or predictive analytics. The use of simulation models in a DT implementation provides additional advantage when its updated parameters reflect the behaviour of an instance of the Physical Twin, and this makes these simulation models useful for offering additional insights to support decision-making, predicting anomalies or future failures [35]. This is especially true for symbiotic simulation models (SSM) because of their ability to interact with physical systems using near real-time or real-time data [2]. SSM differs from non-SSM with its ability to read data at run-time and response according to the current state of the physical system it is representing [23]. Nwogu et al. [22] include SSM within a DT architecture and suggest that SSM can be used to (1) detect anomalies, (2) validate the models in order to identify the most accurate representation, (3) forecast the system behaviour, and (4) control the physical system.

4.3.6. Optimisation Module

Thanks to the capability to fuse disparate data and synchronise with a real-world entity, a DT is able to optimise a system along its entire lifecycle, which includes its design, development, operate and retire stages [14]. Depending on the requirements of a use case, the DT Optimisation Module, in collaboration with the Scenario Manager, may use the output of other DT components (e.g., Data Analytics, Simulation) to improve the performance of its real-world counterpart. This may result in automated control via an actuator or manual control via a user, of the Physical Twin, in which its parameters are modified based on the outcome of the optimisation [23].

4.3.7. Control Module

Once the simulation-optimization cycles are concluded, the scenario manager can collect results and covert them into a set of instructions for the physical system. This set of instructions is analysed by the Control Module, to verify if the action is still applicable to the real-world system and feasible within the rest of the production epoch, in case of a manufacturing system. Once verified, the controls are passed to the actuators, which control the Physical Twin automatically; this is made possible by bidirectional communication between the DT and its Physical Twin [34], and differentiates a DT from other digital forms [12]. Controls from a DT to a Physical Twin can also be exerted manually by a decision-maker or user [34].

5. Conclusions and Further Developments

Several studies have proposed DT architectures that are largely domain- and technology-specific. Although the use cases, applications, and domains may vary, the components of the architectures and the naming of the components differ significantly. As a consequence, it is difficult to identify commonalities in components and this may hinder the re-usability of DT architectures. In this work, a connection is made between the core requirements of a DT and the components of the proposed architecture. These components are named using standard descriptions that depict their functionalities. This makes it easier to apply them to a wide spectrum of applications and use cases. By matching the components of the architecture to the core DT requirements, this paper contributes to the further exploration and standardisation of DT architectures. Also, this work contributes to clarifying the widespread misconception on what constitutes a Digital Twin. This work is affected by several limitations, such as the relationship between the DT requirements and their implementation within an existing information system, additional requirements that may arise from specific manufacturing or service systems, amongst others. Future work should investigate these limitations, as well as validate the suitability and viability of the proposed architecture by applying it to various use cases.

References

[1] Alam, K.M., El Saddik, A., 2017. C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems. IEEE access 5, 2050–2062.

[2] Aydt, H., Turner, S.J., Cai, W., Low, M.Y.H., 2008. Symbiotic simulation systems: An extended definition motivated by symbiosis in biology, in: 2008 22nd Workshop on Principles of Advanced and Distributed Simulation, IEEE. pp. 109–116.

[3] Borangiu, T., Oltean, E., Răileanu, S., Anton, F., Anton, S., Iacob, I., 2019. Embedded digital twin for arti-type control of semi-continuous production processes, in: International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing, Springer. pp. 113–133.

[4] Chaux, J.D., Sanchez-Londono, D., Barbieri, G., 2021. A digital twin architecture to optimize productivity within controlled environment agriculture. Applied Sciences 11, 8875.

[5] Damjanovic-Behrendt, V., Behrendt, W., 2019. An open source approach to the design and implementation of digital twins for smart manufacturing. International Journal of Computer Integrated Manufacturing 32, 366–384.

[6] Dittmann, S., Zhang, P., Glodde, A., Dietrich, F., 2021. Towards a scalable implementation of digital twins-a generic method to acquire shopfloor data. Procedia CIRP 96, 157–162.

[7] Farsi, M., Ariansyah, D., Erkoyuncu, J.A., Harrison, A., 2021. A digital twin architecture for effective product lifecycle cost estimation. Procedia CIRP 100, 506–511.

[8] Ghita, M., Siham, B., Hicham, M., 2020. Digital twins development architectures and deployment technologies: Moroccan use case. International Journal of Advanced Computer Science and Applications (IJACSA) 11.

[9] Glaessgen, E., Stargel, D., 2012. The digital twin paradigm for future nasa and us air force vehicles, in: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA, p. 1818.

[10] Hanisch, A., Tolujew, J., Schulze, T., 2005. Initialization of online simulation models, in: Proceedings of the Winter Simulation Conference, 2005., IEEE. pp. 9–pp.

[11] Josifovska, K., Yigitbas, E., Engels, G., 2019. Reference framework for digital twins within cyber-physical systems, in: 2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS), IEEE. pp. 25–31.

[12] Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. Digital twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 51, 1016–1022.

[13] Lee, J., Bagheri, B., Kao, H.A., 2015. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing letters 3, 18–23.

[14] Leng, J., Jiang, P., 2019. Dynamic scheduling in rfid-driven discrete manufacturing system by using multi-layer network metrics as heuristic information. Journal of Intelligent Manufacturing 30, 979–994.

[15] Leng, J., Zhou, M., Xiao, Y., Zhang, H., Liu, Q., Shen, W., Su, Q., Li, L., 2021. Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. Journal of Cleaner Production 306, 127278.

[16] Liu, M., Fang, S., Dong, H., Xu, C., 2021. Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems 58, 346–361.

[17] Lugaresi, G., Matta, A., 2018. Real-time simulation in manufacturing systems: Challenges and research directions, in: 2018 Winter Simulation Conference (WSC), IEEE. pp. 3319–3330.

[18] Lugaresi, G., Matta, A., 2021. Automated manufacturing system discovery and digital twin generation. Journal of Manufacturing Systems 59, 51–66. doi:https://doi.org/10.1016/j.jmsy.2021.01.005.

[19] Mourtzis, D., Togias, T., Angelopoulos, J., Stavropoulos, P., 2021. A digital twin architecture for monitoring and optimization of fused deposition modeling processes. Procedia CIRP 103, 97–102.

[20] Moyne, J., Qamsane, Y., Balta, E.C., Kovalenko, I., Faris, J., Barton, K., Tilbury, D.M., 2020. A requirements driven digital twin framework: Specification and opportunities. IEEE Access 8, 107781–107801.

[21] Mustafee, N., Powell, J., Harper, A., 2018. Right hospital-right time: A business analytics framework for analysing urgent care/a&e wait time data, in: Proceedings of the 2018 Simulation Workshop, The Operational Research Society. pp. 105–115.

[22] Nwogu, C., Taylor, S.J.E., Anagnostou, A., . Towards a Generic Architecture for Symbiotic Simulation System-based Digital Twin. Proceedings of the 2020 Winter Simulation Conference, December 2020, pp. 365. URL: https://informs-sim.org/wsc20papers/365.pdf.

[23] Onggo, B.S., Juan, A.A., Mustafee, N., Smart, A., Molloy, O., Juan, A.A., Molloy, O., 2018. Symbiotic simulation system: hybrid systems model meets big data analytics, in: 2018 Winter Simulation Conference (WSC), IEEE. Institute of Electrical and Electronics Engineers Inc.. pp. 1358–1369. doi:10.1109/WSC.2018.8632407.

[24] Qi, Q., Tao, F., Zuo, Y., Zhao, D., 2018. Digital twin service towards smart manufacturing. Procedia Cirp 72, 237–242.

[25] Redelinghuys, A., Basson, A.H., Kruger, K., 2020. A six-layer architecture for the digital twin: a manufacturing case study implementation. Journal of Intelligent Manufacturing 31, 1383–1402.

[26] Rhodes-Leader, L., Onggo, B.S.S., Worthington, D.J., Nelson, B.L., 2018. Airline disruption recovery using symbiotic simulation and multi-fidelity modelling .

[27] Robinson, S., 2014. Simulation: the practice of model development and use. Bloomsbury Publishing.

[28] Rosen, R., Von Wichert, G., Lo, G., Bettenhausen, K.D., 2015. About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine 48, 567–572.

[29] Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., Wang, L., 2012. Modeling, simulation, information technology & processing roadmap. National Aeronautics and Space Administration 32, 1–38.

[30] Souza, V., Cruz, R., Silva, W., Lins, S., Lucena, V., 2019. A digital twin architecture based on the industrial internet of things technologies, in: 2019 IEEE International Conference on Consumer Electronics (ICCE), IEEE. pp. 1–2.

[31] Steindl, G., Stagl, M., Kasper, L., Kastner, W., Hofmann, R., 2020. Generic digital twin architecture for industrial energy systems. Applied Sciences 10, 8903.

[32] Tao, F., Qi, Q., Liu, A., Kusiak, A., 2018. Data-driven smart manufacturing. Journal of Manufacturing Systems 48, 157–169.

[33] Uhlemann, T.H.J., Schock, C., Lehmann, C., Freiberger, S., Steinhilper, R., 2017. The digital twin: demonstrating the potential of real time data acquisition in production systems. Procedia Manufacturing 9, 113–120.

[34] van der Valk, H., Haße, H., Möller, F., Otto, B., 2021. Archetypes of digital twins. Business & Information Systems Engineering , 1–17.

[35] VanDerHorn, E., Mahadevan, S., 2021. Digital twin: Generalization, characterization and implementation. Decision Support Systems , 113524.

[36] Vrabič, R., Erkoyuncu, J., Farsi, M., Ariansyah, D., 2021. An intelligent agent-based architecture for resilient digital twins in manufacturing. CIRP annals 70, 349–352.