Conceptualization of a Platform Architecture for the Rapid Development of Digital Twins in Manufacturing

Luis Felipe Villegas, Marco Macchi, Adalberto Polenghi

Department of Management, Economics, and Industrial Engineering,
Politecnico di Milano, Via Lambruschini 4/B, 20156, Milan, Italy (luisfelipe.villegas@polimi.it,
marco.macchi@polimi.it, adalberto.polenghi@polimi.it)

Abstract: Digital Twins (DTs) have emerged as promising systems within manufacturing, enabling capabilities such as real-time monitoring, predictive maintenance, and optimization of products and processes. Their development still faces significant challenges, including the need for rapid prototyping, scalability, and interoperability. This paper aims to propose a Digital Twin Platform (DTP) architecture conceptualized in order to address these challenges. The DTP builds on a combined approach of composability, low-code, and collaborative methodologies for accelerated Digital Twin (DT) development. Particularly, by envisioning a composable approach for developing, deploying, and managing DTs across manufacturing environments, the architecture aims to set the basis of a platform to enable rapidly create highly adaptable, scalable and interoperable DT solutions fitted to specific manufacturing requirements. Building on an explorative literature review, the proposed architecture is conceptualized in five environments; each environment tackles specific aspects of the DT lifecycle management, utilizing low-code development, composable services and collaboration for faster development and integration. The paper also exemplifies the use of the proposed architecture by showcasing its application within a use-case selected from academic literature. Overall, the proposed architecture intends to lay the groundwork for a reference platform that empowers researchers and practitioners to accelerate the development, customization, and integration of DTs, with the ultimate goal to unlock the achievements of the potential benefits of DTs for competitive and sustainable manufacturing.

Keywords: Digital Twin, platform, architecture, composability, manufacturing.

1. Introduction

Digital Twins (DTs) are complex engineering systems (Zhang, Zhou and Horn, 2021) developed to offer decision support capabilities by leveraging models and data collected from both the physical and virtual space that conform them (Kritzinger *et al.*, 2018). DTs are a key part of shaping the path towards the new smart era of manufacturing. Particularly, DTs are promising in order to provide manufacturers with valuable insights into the performance and health of their products, assets, production processes and services. These complex systems can be used to improve their efficiency, reduce downtime, and optimize maintenance schedules among other applications, see for example (Kritzinger *et al.*, 2018; Barricelli, Casiraghi and Fogli, 2019; Cimino, Negri and Fumagalli, 2019).

Despite the potential benefits of DTs, there are several challenges associated with their development and deployment. These challenges range from rapid prototyping (Trauer *et al.*, 2022) to scalability (Michael and Wortmann, 2021; Qamsane *et al.*, 2021) and interoperability (Michael and Wortmann, 2021; Cimino *et al.*, 2023; Kim *et al.*, 2023; Karabulut *et al.*, 2024). Overall, these challenges still need to be addressed in order to fully unlock the potential benefits of DTs across diverse manufacturing domains. To this end, this paper proposes the concept of a Digital Twin Platform (DTP) architecture tailored to enable the rapid DT development and deployment, and its related management processes. The DTP adopts the concept of composability, as a way to aggregate DT services on a single DT solution, low-code development, to democratize the development of DTs, and collaborative methodologies, as the main levers. As such, the DTP architecture seeks to provide developers with the tools to rapidly create highly adaptable, scalable and interoperable DT solutions, customized to meet specific manufacturing requirements.

Considering this as the main objective, a research question is stated as the major driver: "*How rapid development and deployment of DTs can be enabled to meet the requirements of different manufacturing settings?*". Driven by this question, the ultimate goal is to set the foundations for a reference platform that should enable to accelerate the development, customization, and integration of DTs in real manufacturing settings. By doing so, this research contributes to unlock the potential of DTs to achieve the different benefits for a competitive and sustainable manufacturing.

The complex nature of the research question requires different activities during the investigation, and this paper only reports the outcome of an initial activity. Particularly, an explorative literature review was conducted, focusing on key areas including the challenges in DT development and deployment, the efficacy of low-code development and collaborative methodologies in order to leverage composability, and the benefits of a composable approach as the key driver. Therefore, drawing upon the insights from the literature review, the architecture consists of five environments, each one proposed in order to address specific facets of the DT development lifecycle (Moyne *et* *al.*, 2020; Niu *et al.*, 2023). Indeed, leveraging the principles of low-code development, composability and collaboration, these environments streamline the DT development process, with the end purpose to facilitate the rapid DT development and deployment, in order to finally achieve seamless integration in different manufacturing settings.

A practical application of the proposed DTP architecture is also described, built on a use-case selected from the academic literature. The use case is adopted to showcase the application of the architecture: through this example, the paper describes how current DT development methodologies could be fitted into the proposed methodologies. This aims to provide an initial evidence of the architecture's feasibility, effectiveness, and potential impact on the DT development lifecycle within the manufacturing domain.

The paper is correspondingly organized. The explorative literature review is firstly presented in a brief overview, both including the methodology and literature findings (section 2). The proposed DTP architecture is outlined, building on such findings (section 3), and then showcasing its application with the use case selected from the literature (section 4). Finally, the conclusions (section 5) report a synthesis of the main achievements and their expected follow-ups activities to give continuity to the research.

2. Methodology and Findings

2.1. Methodology

An explorative literature review was conducted in order to identify the key characteristics needed from a DT platform (DTP) architecture, to be able to address the rapid development, scalability and interoperability challenges associated with the development of DTs. This non-systematic review process entailed the identification of pertinent scientific papers within the domains of DTs, including software engineering, and manufacturing. A methodological selection criterion was employed to ensure the inclusion of literature that directly contributed to the understanding and advancement of DTPs and architectures. In total, 22 papers were included based on their relevance and significance to the research topic.

To form this corpus of literature, an explorative search was conducted across various databases, including Scopus, IEEE Xplore, and Web of Science. The inclusion criteria for the literature review were specifically defined to ensure the selection of papers that best aligned with the objective of the research. Particularly, preference was given to papers, written in English, that specifically addressed DTPs or architectures, identified challenges pertinent to the DT development and deployment, and proposed innovative solutions to overcome these challenges. Moreover, papers published after 2018 were then prioritized to incorporate the latest advancements and perspectives in the field.

Upon the collection and review of the selected papers, a systematic analysis was undertaken to identify key concepts, features, and platform architectures and delineate prominent challenges within the realm of DT development. This comprehensive analysis not only facilitated a deeper understanding of the complexities inherent in DT systems but also served as a crucial source of insights for detailing the conceptual design of the proposed DTP architecture.

2.2. Findings

The literature findings are arranged in the following key areas: i) challenges and requirements in DT development and deployment, ii) main DT architectures, DTP or models to support the stages of the DT development lifecycle, and iii) innovative solutions for DT development, including the efficiency of low-code development, collaborative methodologies, and the benefits of a composable approach as a key driver. Such perspectives are elaborated in the reminder of this section, with **Table 1** summarizing the findings.

Table 1: classification of the literature findings

Publication	Objective	Class. Area
(Silva *et al.*, 2023)	Digital twin platform to enhance power transformer performance and lifespan.	ii)
(Qamsane *et al.*, 2021)	System Development Lifecycle-based methodology for developing DT solutions in manufacturing.	ii)
(Moyne *et al.*, 2020)	Baseline framework for smart manufacturing with DT.	ii)
(van Schalkwyk and Isaacs, 2023)	Introducing Composable and Lean DTs for industrial organizations.	ii), iii)
(Li *et al.*, 2021)	Blockchain-based DT sharing platform for efficient resource integration.	ii)
(Trauer *et al.*, 2022)	Survey to identify challenges of DT implementation in industry.	i)
(Kim *et al.*, 2023)	Customized DT platform to support SME competitiveness.	ii)
(Bononi *et al.*, 2023)	Applications of DTs in various fields and collaboration technologies.	ii), iii)
(Leng *et al.*, 2023)	DT-driven reconfigurable manufacturing systems for enhanced efficiency.	ii)
(Zhang *et al.*, 2024)	Blockchain and DT-based platform for production logistics.	ii)
(Hasidi *et al.*, 2023)	Scalable DT architecture for mineral processing.	ii)

(Niu *et al.*, 2023)	DTP to facilitate SMEs' transition to service-centered business models.	i), ii)
(Cimino *et al.*, 2023)	Multi-purpose DTP to optimize SME production processes and products, ensuring interoperability.	ii)
(Bellavista and Di Modica, 2024)	Design of an open industrial DTP.	ii)
(Karabulut *et al.*, 2024)	Literature review on ontologies in DTs across domains.	ii)
(Tripathi *et al.*, 2024)	Standardization in DT ecosystems for stakeholder collaboration.	i), ii)
(Kasper *et al.*, 2022)	Tailored DTP for industrial energy systems.	i), ii)
(Redeker *et al.*, 2021)	Role of DTs and DTPs in manufacturing productivity challenges.	i), ii)
(Dalibor *et al.*, 2020)	Model-driven approach for interactive DTs in smart manufacturing.	i), ii)
(Ogunsakin, Mehandjiev and Marin, 2023)	Feasibility of online optimization in DTs for mass personalization.	ii)
(Michael and Wortmann, 2021)	Adaptable DTs in smart manufacturing for efficient production system operation.	i), ii), iii)
(Borodulin *et al.*, 2017)	Platform for the creation and deployment of DTs of industrial processes and systems.	ii), iii)

3. The Platform Architecture

The proposed DTP architecture seeks to provide a holistic view of the general aspects to be handled along the DT development lifecycle. Derived from the literature findings, this architecture integrates the main components of DT and DTP architectures, such as data, model, service and version management, and user interaction, and incorporates methodologies for DT development, including agile development, low-code platforms, and collaborative frameworks. As illustrated in **Figure 1**, the architecture is structured into five environments, or layers, each serving a distinct purpose in facilitating the design, development, deployment, and security management of DTs.

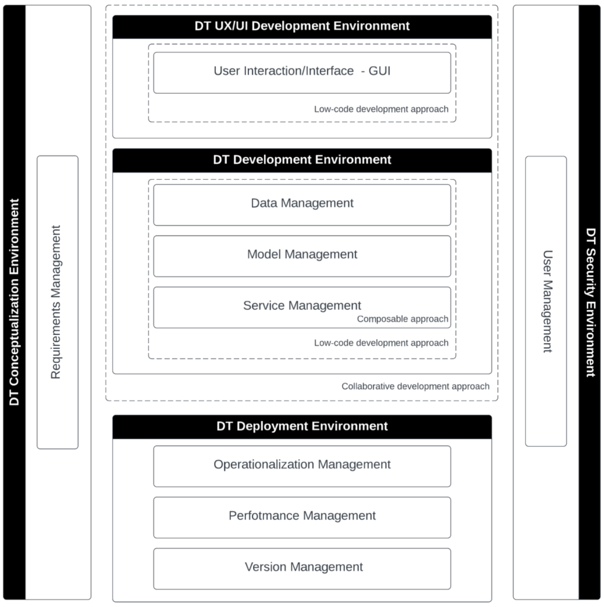

Figure 1: proposed Digital Twin Platform architecture.

This structured approach aims to allow DT developers to fine-tune specific aspects within each layer, ensuring the resulting DT effectively meets the desired functionalities and requirements of the DT user. Therefore, by synthesizing the key elements identified in the literature, the proposed architecture aims to address the rapid development, scalability, and interoperability challenges associated with DTs, thereby enhancing the overall efficiency and effectiveness of the DT development process.

3.1. DT Conceptualization Environment

As stated by (Borodulin *et al.*, 2017; Kasper *et al.*, 2022; Bononi *et al.*, 2023; Silva *et al.*, 2023), the development process of DTs needs the capacity to effectively manage their requirements. To address this necessity, the *DT Conceptualization Environment* envisions a collaborative space for the key stakeholders (e.g., engineers, production managers, data scientists) to define and manage the requirements for the DT to be developed. As a supporting methodology, the Agile methodology aligns well with the dynamic and evolving nature of DT development projects, facilitating an iterative requirement gathering, refinement and management (Hernández, Moros and Nicolás, 2023). In the frame of an Agile methodology, the key stakeholders can use tools such as, e.g., user story maps and backlog prioritization to define the functionalities and features of the DT. In the end, the DT developers can then use these inputs to ensure that the development of the DT aligns closely with the expectations and objectives set by the key stakeholders.

Since this environment acts as the foundation of a DT project, ensuring all development activities are aligned, the proposed DTP architecture places it prominently on the far-left side of the framework (**Figure 1**). This visually emphasizes how the other environments should be aligned with the principles established in this initial environment.

3.2. DT UX/UI Development Environment

This environment specifically focuses on the creation of the User Interaction/Interface (UI) through an intuitive and engaging user experience (UX) Graphical User Interface (GUI), also called DT cockpits, in accordance with (Dalibor *et al.*, 2020; Michael and Wortmann, 2021; Hasidi *et al.*, 2023; Kim *et al.*, 2023; Silva *et al.*, 2023; van Schalkwyk and Isaacs, 2023). More in detail, utilizing a low-code development approach (Michael and Wortmann, 2021) and fostering a collaborative practice among the DT developers (Bononi *et al.*, 2023), this environment aims to enable the seamless design and customization of the DT's interface. The design of the UI must adhere to the specifications outlined in the *DT Conceptualization Environment*, thus ensuring coherence and effectiveness in the user interaction.

This environment is located in the top-centre part of the framework (**Figure 1**) underscoring the relevance of the influence of the DT user on the DT development lifecycle.

3.3. DT Development Environment

The *DT Development Environment* is the core of the proposed architecture. It is responsible for creating the DT services that provide the DT's functionality. This environment is divided into three sub-layers:

- Data Management: Following the proposed architectures by (Dalibor *et al.*, 2020; Li *et al.*, 2021; Michael and Wortmann, 2021; Redeker *et al.*, 2021; Kasper *et al.*, 2022; Bononi *et al.*, 2023; Hasidi *et al.*, 2023; Kim *et al.*, 2023; Leng *et al.*, 2023; Silva *et al.*, 2023; van Schalkwyk and Isaacs, 2023; Bellavista and Di Modica, 2024; Tripathi *et al.*, 2024; Zhang *et al.*, 2024), this sub-layer is used to manage and integrate all of the data sources that will be consumed by the DT. Data sources can include data from sensors, machines, enterprise resource planning (ERP) systems, knowledge graphs, and other data repositories. Within this layer, the concept of a data lake emerges as a main component, offering a centralized repository to consolidate and organize heterogeneous data streams. The *Data Management* sub-layer envisions the presence of tools for different needs such as data cleansing, transformation, and integration, following the low-code development approach for its practical configuration.
- Model Management: This sub-layer is intended to be used as a hub to manage the models that the DT will use (Li *et al.*, 2021; Michael and Wortmann, 2021; Kasper *et al.*, 2022; Kim *et al.*, 2023; Silva *et al.*, 2023; van Schalkwyk and Isaacs, 2023; Zhang *et al.*, 2024). These models can include data models, simulation models, machine learning models, etc. Moreover, the *Model Management* sub-layer aims to provide a library of reusable models that can be adopted to rapidly create new DTs. The low-code development approach is required to easily drag and drop model from different nature and combine them to make a unique complex model.
- Service Management: This sub-layer seeks the creation of specific DT services or applications (Michael and Wortmann, 2021; Redeker *et al.*, 2021; Kasper *et al.*, 2022; Bononi *et al.*, 2023; Kim *et al.*, 2023; Leng *et al.*, 2023; Silva *et al.*, 2023; Bellavista and Di Modica, 2024; Karabulut *et al.*, 2024; Zhang *et al.*, 2024), such as product maintenance, production optimization, risk management, etc. Services are created by combining data from the *Data Management*, models from the *Model Management*, and additional building blocks to construct the desired logic of the service to be provided. These building blocks can include functionalities such as data analytics, machine learning algorithms, calls to simulation software, rule-based decision-making, and optimization models. To foster composability, the *Service Management* sub-layer utilizes a low-code development approach to facilitate service creation, allowing DT developers to assemble services from pre-built components, without extensive coding.

  A key aspect of them is their composable nature (van Schalkwyk and Isaacs, 2023). Services are in fact designed to be independent and reusable. This means that services can be easily combined and recombined to create new decision capabilities from DTs (Niu *et al.*, 2023). This composable approach fosters rapid development and reduces development costs.

On the whole, a collaborative approach is also fostered in the *DT Development Environment*: DT developers are typically organized in teams of specialists which are dealing with the different facets within and across the sub-layers of *Data Management*, *Model Management* and *Service Management*.

This environment is situated on the middle-centre of the architecture (**Figure 1**) to underscore its pivotal role as the core engine driving the development of the DT's capabilities and functionalities, placed on a layer below *DT UX/UI Development Environment* to be hidden from the DT user and exposed to the DT developer.

3.4. DT Deployment Environment

This environment is envisioned to operate and manage the developed DTs. It is divided into three main sub-layers:

- Operationalization Management: This sub-layer seeks to manage the deployment and configuration of the DTs (Moyne *et al.*, 2020; Li *et al.*, 2021; Redeker *et al.*, 2021; Niu *et al.*, 2023; Silva *et al.*, 2023; Bellavista and Di Modica, 2024). It should provide tools for managing DT instances, scaling DTs to meet demand, and integrating DTs with other systems. Integration is a key issue and should build upon the interoperability of the DTs with the manufacturing settings where they are introduced.

- Performance Management: This sub-layer intends to be used to monitor the performance of the deployed DTs and identify any issues such as drifts (Moyne *et al.*, 2020; Bellavista and Di Modica, 2024). Configured with real-time dashboards and alerts, it aims to allow operators to monitor the status and performance of DTs.
- Version Management: This sub-layer is used to manage different versions of DTs (Moyne *et al.*, 2020; Redeker *et al.*, 2021). It allows DT developers to track changes to DTs, roll back to previous versions if necessary, and manage the lifecycle of DTs as subsequent developments.

It is positioned at the bottom-centre of the architecture (**Figure 1**) to remark its critical role in translating the DT from development to operational phases seamlessly, by being on a layer below *DT Development Environment*.

3.5. DT Security Environment

The *DT Security Environment*, as the final layer, is proposed as a management layer responsible for the DT security and the data it contains. As such, this layer should provide all features such as user authentication and authorization, data encryption, and access control (van Schalkwyk and Isaacs, 2023; Bellavista and Di Modica, 2024). This environment aims to ensure that only authorized users can access certain features of the deployed DT by setting up proper user roles with specific credentials.

This environment is placed on the far-right side of the architecture (**Figure 1**) on the one hand, to address its influence on each of the other environments and on the other hand, to emphasize its role as a protective barrier against potential external security threats.

4. Application of the Platform Architecture

The present section showcases how the proposed DTP architecture can be utilized in order to implement a DT solution, drawing insights from the use case presented in "Asset Administration Shell as an interoperable enabler of Industry 4.0 software architectures: a case study" by (Quadrini *et al.*, 2023). The use case outlined in this article focuses on developing a digital model of a production line, in a laboratory environment (at the Industry4.0Lab), using Asset Administration Shell (AAS), a standardized framework for managing and exchanging data of industrial assets, to ensure interoperability and standardization in data management of the line. This article is part of the work of researchers within the same research group at the Politecnico di Milano, i.e. the Manufacturing Group. Therefore, the authors of the present article have detailed information about the way the digital model of the production line is implemented.

The AAS is a key method, used to represent the production line with modular metamodels, which are then adopted in order to feed a digital model representing the line configuration. This approach proves effective for virtual commissioning tasks, demonstrating the potential of the AAS as an enabler of Industry 4.0 software architectures, according to the original intended work by (Quadrini *et al.*, 2023). Hereafter, such approach is extended, exploring how the proposed DTP architecture can be leveraged in order to implement a DT solution maintaining the AAS method presented in the original use case. It is also discussed how the architecture's five environments can be further utilized to enhance, by means of a layered structure, the entire development process, inclusive of the design, development, deployment, and security management of the developed DT.

On the whole, the perspective of this showcase is the one of an industrial engineer. Indeed, industrial engineers play a crucial role in leveraging innovative technologies in order to optimize the industrial processes and to enhance overall operational efficiency. Thus, they should be habilitated to attend the entire development process, in coherence with their role in industrial process innovation.

DT Conceptualization Environment: In this initial phase, industrial engineers are typically involved as they would be collaborating with the key stakeholders in order to define the requirements and the objectives of the DT solution. Following the Agile methodology, they would iteratively refine the conceptualization of the DT, ensuring alignment with the overarching goals of the production line digitalization project. Key considerations would include identifying critical data sources, defining the performance metrics of interest, and outlining the desired functionalities of the DT. It should be noted that, for the matter of the article where the use case is originally presented, such functionalities are limited to the DT use as a virtual commissioning system to save time on the set-up of the line.

DT UX/UI Development Environment: With the requirements clearly defined, DT developers would proceed to design the user UI for interacting with the DT. Leveraging the low-code development approach and fostering collaborative efforts, DT developers would create an intuitive GUI that enables users to interact with the digital model of the production line effectively. For the specific needs of the use case according to what was originally presented, the authors would have only needed to drop a simple visualization frame and leverage from the CAD drawings of the stations, provided by the utilized software, in order to monitor the production line. Finally, it is worth noting that this layer implies the interaction of DT developers with industrial engineers involved at first layer of the DTP architecture; potentially, thanks to the low-code development, industrial engineers may have sufficient skills also to configure some GUI according to the needs.

DT Development Environment: within the *Data Management* sub-layer, DT developers would integrate data from various sources, including sensors, PLCs, and MES, into a unified data repository. Building on what presented in the original article, the authors at this point would require to include general connectors to their SHIELD architecture, taking advantage of the implemented OPC UA data exchange protocol, in order to be able to collect and store the data from the line into the DTP. Thereafter, coherently with the prior article, at the *Model Management* sub-layer, the authors could build upon the modular metamodels defined in the

AAS in order to characterize production line's behaviour and performance. At this sub-layer, they would also store and manage the CAD drawings that are used within the *DT UX/UI Development Environment*. Any other additional data or models could be included at this sub-layer anticipating other future requirements or services. Finally, within the *Service Management* sub-layer, DT developers would create specific DT services to be provided by the DTP. Although the analysed article originally presents only a limited service of the DT in relationship to the tasks required as a virtual commissioning system, at this point authors may leverage the composable approach to create multiple DT services tailored to the needs of the production line. This should depend on the requirements that are established in the *DT Conceptualization Environment*, and may lead to multiple DT services related for example to the predictive maintenance, production performance monitoring, and/or production scheduling optimisation. As final remark, it is worth noting that this layer implies the key role of DT developers, as they may be able to better fit data, model, service management to the purpose. The low-code development may facilitate industrial engineers in their involvement, even if skills are much higher, due to the spread of requirements across data, models and services.

DT Deployment Environment: There is no further work from the authors of the original article up to this point, although once the DT solution is developed into the production environment, the DTP, at the *Operationalization Management* sub-layer, could allow them to have a dedicated space to ensure its seamless integration with existing systems and infrastructure from the laboratory. On the *Performance Management,* they would be enabled to monitor and optimize the performance of the deployed DT, fine-tuning parameters and configurations to enhance its efficiency. At the *Version Management* sub-layer authors would be allowed to manage iterative improvements, adding new services to the current virtual commissioning system, and updates to accommodate the evolving needs while ensuring its backward compatibility, when the system requires to get downgraded.

DT Security Environment: Finally, in this environment, robust security measures for specific features of the DT solution should be implemented. For the matter of the provided solution by the authors of the prior article, they should assign unique user roles, that limit or enable the visualization or modification of specific data and parameters. For instance, a System Administrator role would be tasked with the overall management and administration of the DT system, including user access control, system configuration, and related service tasks. An Engineer/Designer role would be involved in designing and configuring the DT's models, defining simulation scenarios, and optimizing system performance, having access to advanced features and tools for model development. Meanwhile, an Operator role would be responsible for operating the virtual commissioning environment, running simulations, and analysing simulation results, thus having partial access to the backend data of the DT system.

5. Conclusions

The proposed DTP architecture offers a comprehensive framework for addressing the challenges associated with the rapid development, deployment, and management of DTs within the manufacturing domain. By leveraging a combination of composability, low-code development, and collaborative methodologies, the architecture aims to enable DT developers and stakeholders to create highly adaptable, scalable and interoperable DT solutions that address specific manufacturing needs.

The five distinct environments within the DTP architecture facilitate a structured and streamlined approach to the DT development lifecycle. This allows the detailed customization of each aspect of the DT, ensuring optimal functionality and alignment with user requirements. The low-code development approach embedded within the architecture aims to significantly reduce development time and effort. As seen within the academic literature, this democratizes DT development, allowing a wider range of stakeholders, including those with limited coding expertise, to actively participate in the creation and customization of DTs. The composable nature of the services within the architecture is another key strength. By intending to enable the seamless combination and recombination of pre-built service blocks, the architecture promotes rapid prototyping and facilitates the efficient and quick creation of complex DT functionalities. This not only might help reduce development costs but also increase the rate of innovation and experimentation within the DT development process.

The integration of insights from the explorative literature review and the description of its practical application, within an academic use case, underscores the architecture's feasibility and potential impact. Although the proposed architecture has the potential to significantly accelerate the adoption and utilization of DTs across diverse manufacturing domains, the continuous validation and refinement of the DTP architecture will be crucial to enhance its adaptability and support for real-world DT implementations, further unlocking the transformative benefits of DTs in competitive and sustainable manufacturing environments. In this matter, the proposed DTP architecture lays a solid foundation for further research and development efforts. Future work could start by setting a beta version of a DTP based on the proposed architecture to practically validate its functionality. Further work would focus on exploring the integration of advanced functionalities such as machine learning and artificial intelligence within the architecture. Additionally, research into the development of standardized DT libraries and reusable components would further enhance the composability and efficiency of the architecture. Finally, the implementation and validation of the DTP architecture across a wider range of real-world manufacturing use cases will provide valuable insights into its practical effectiveness and potential for broader industry adoption.

References

Barricelli, B.R., Casiraghi, E. and Fogli, D. (2019) 'A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications’, *IEEE Access*, 7, pp. 167653–167671. Available at: https://doi.org/10.1109/ACCESS.2019.2953499.

Bellavista, P. and Di Modica, G. (2024) ‘IoTwins: Implementing Distributed and Hybrid Digital Twins in Industrial Manufacturing and Facility Management Settings’, *Future Internet*, 16(2), p. 65. Available at: https://doi.org/10.3390/fi16020065.

Bononi, L. *et al.* (2023) ‘Digital Twin Collaborative Platforms: Applications to Humans-in-the-Loop Crafting of Urban Areas’, *IEEE Consumer Electronics Magazine*, 12(6), pp. 38–46. Available at: https://doi.org/10.1109/MCE.2022.3214944.

Borodulin, K. *et al.* (2017) ‘Towards Digital Twins Cloud Platform: Microservices and Computational Workflows to Rule a Smart Factory’, in *Proceedings of the10th International Conference on Utility and Cloud Computing. UCC ’17: 10th International Conference on Utility and Cloud Computing*, Austin Texas USA: ACM, pp. 209–210. Available at: https://doi.org/10.1145/3147213.3149234.

Cimino, A. *et al.* (2023) ‘Integrating multiple industry 4.0 approaches and tools in an interoperable platform for manufacturing SMEs’, *Computers & Industrial Engineering*, 186, p. 109732. Available at: https://doi.org/10.1016/j.cie.2023.109732.

Cimino, C., Negri, E. and Fumagalli, L. (2019) ‘Review of Digital Twin Applications in Manufacturing’, *Computers in industry*, 113, p. 103130.

Dalibor, M. *et al.* (2020) ‘Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits’, in G. Dobbie et al. (eds) *Conceptual Modeling*. Cham: Springer International Publishing, pp. 377–387. Available at: https://doi.org/10.1007/978-3-030-62522-1_28.

Hasidi, O. *et al.* (2023) ‘Generic and scalable multi-layered architecture for Digital Twin implementation in industrial processes: Mineral Processing case study’, in *Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence. ICAAI 2023: 2023 the 7th International Conference on Advances in Artificial Intelligence*, Istanbul Turkiye: ACM, pp. 54–61. Available at: https://doi.org/10.1145/3633598.3633622.

Hernández, R., Moros, B. and Nicolás, J. (2023) ‘Requirements management in DevOps environments: a multivocal mapping study’, *Requirements Engineering*, 28(3), pp. 317–346. Available at: https://doi.org/10.1007/s00766-023-00396-w.

Karabulut, E. *et al.* (2024) ‘Ontologies in digital twins: A systematic literature review’, *Future Generation Computer Systems*, 153, pp. 442–456. Available at: https://doi.org/10.1016/j.future.2023.12.013.

Kasper, L. *et al.* (2022) ‘Toward a Practical Digital Twin Platform Tailored to the Requirements of Industrial Energy Systems’, *Applied Sciences*, 12(14), p. 6981. Available at: https://doi.org/10.3390/app12146981.

Kim, G.-Y. *et al.* (2023) ‘Customized Digital Twin Platform for SMEs in South Korea’, *IFAC-PapersOnLine*, 56(2), pp. 11044–11049. Available at: https://doi.org/10.1016/j.ifacol.2023.10.806.

Kritzinger, W. *et al.* (2018) ‘Digital Twin in Manufacturing: A Categorical Literature Review and Classification’, *IFAC-PapersOnLine*, 51(11), pp. 1016–1022. Available at: https://doi.org/10.1016/j.ifacol.2018.08.474.

Leng, B. *et al.* (2023) ‘Digital twin monitoring and simulation integrated platform for reconfigurable manufacturing systems’, *Advanced Engineering Informatics*, 58, p. 102141. Available at: https://doi.org/10.1016/j.aei.2023.102141.

Li, M. *et al.* (2021) ‘Blockchain-based digital twin sharing platform for reconfigurable socialized manufacturing resource integration’, *International Journal of Production Economics*, 240, p. 108223. Available at: https://doi.org/10.1016/j.ijpe.2021.108223.

Michael, J. and Wortmann, A. (2021) ‘Towards Development Platforms for Digital Twins: A Model-Driven Low-Code Approach’, in. *IFIP International Conference on Advances in Production Management Systems (APMS)*, Springer International Publishing, p. 333. Available at: https://doi.org/10.1007/978-3-030-85874-2_35.

Moyne, J. *et al.* (2020) ‘A Requirements Driven Digital Twin Framework: Specification and Opportunities’, *IEEE Access*, 8, pp. 107781–107801. Available at: https://doi.org/10.1109/ACCESS.2020.3000437.

Niu, X. *et al.* (2023) ‘Generic Digital Twin Platform Framework for Advanced Service Innovation’, in *2023 9th International Conference on Virtual Reality (ICVR). 2023 9th International Conference on Virtual Reality (ICVR)*, Xianyang, China: IEEE, pp. 304–308. Available at: https://doi.org/10.1109/ICVR57957.2023.10169843.

Ogunsakin, R., Mehandjiev, N. and Marin, C.A. (2023) ‘Towards adaptive digital twins architecture’, *Computers in Industry*, 149, p. 103920. Available at: https://doi.org/10.1016/j.compind.2023.103920.

Qamsane, Y. *et al.* (2021) ‘A Methodology to Develop and Implement Digital Twin Solutions for Manufacturing Systems’, *IEEE Access*, 9, pp. 44247–44265. Available at: https://doi.org/10.1109/ACCESS.2021.3065971.

Quadrini, W. *et al.* (2023) ‘Asset Administration Shell as an interoperable enabler of Industry 4.0 software architectures: a case study’, *Procedia Computer Science*, 217, pp. 1794–1802. Available at: https://doi.org/10.1016/j.procs.2022.12.379.

Redeker, M. *et al.* (2021) ‘Towards a Digital Twin Platform for Industrie 4.0’, in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS). 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 39–46. Available at: https://doi.org/10.1109/ICPS49255.2021.9468204.

van Schalkwyk, P. and Isaacs, D. (2023) 'Achieving Scale Through Composable and Lean Digital Twins', in N. Crespi, A.T. Drobot, and R. Minerva (eds) *The Digital Twin*. Cham: Springer International Publishing, pp. 153–180. Available at: https://doi.org/10.1007/978-3-031-21343-4_6.

Silva, H. *et al.* (2023) 'A Digital Twin Platform-Based Approach to Product Lifecycle Management: Towards a Transformer 4.0', in J. Machado et al. (eds) *Innovations in Industrial Engineering II*. Cham: Springer International Publishing (Lecture Notes in Mechanical Engineering), pp. 14–25. Available at: https://doi.org/10.1007/978-3-031-09360-9_2.

Trauer, J. *et al.* (2022) 'Challenges in Implementing Digital Twins – a Survey', in. *ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection. Available at: https://doi.org/10.1115/DETC2022-88786.

Tripathi, N. *et al.* (2024) 'Stakeholders collaborations, challenges and emerging concepts in digital twin ecosystems', *Information and Software Technology*, 169, p. 107424. Available at: https://doi.org/10.1016/j.infsof.2024.107424.

Zhang, L., Zhou, L. and Horn, B.K.P. (2021) 'Building a Right Digital Twin with Model Engineering', *Journal of Manufacturing Systems*, 59, pp. 151–164.

Zhang, Z. *et al.* (2024) 'Enhancing trusted synchronization in open production logistics: A platform framework integrating blockchain and digital twin under social manufacturing', *Advanced Engineering Informatics*, 61, p. 102404. Available at: https://doi.org/10.1016/j.aei.2024.102404.