Multi-Partner Project: A Model-Driven Engineering Framework for Federated Digital Twins of Industrial Systems (MATISSE)

Alessio Bucaioni*, Romina Eramo†, Luca Berardinelli‡, Hugo Bruneliere§, Benoit Combemale¶, Djamel Eddine Khelladi‖, Vittoriano Muttillo†, Andrey Sadovykh**, and Manuel Wimmer‡
*Mälardalen Univeristy, Västerås, Sweden
Email: {firstname}.{lastname}@mdu.se
†University of Teramo, Teramo, Italy
‡Johannes Kepler University, Linz, Austria
§IMT Atlantique & LS2N (CNRS), Nantes, France
¶ University of Rennes, France
‖ CNRS, University of Rennes, France
** Softeam, Paris, France

***Abstract*—Digital twins are virtual representations of real-world entities or systems. Their primary goal is to help organizations understand and predict the behaviour and properties of these entities or systems. Additionally, digital twins enhance activities such as monitoring, verification, validation, and testing. However, the inherent complexity of digital twins implies challenges throughout the systems engineering process. This notably includes design, development, and analysis phases, as well as deployment, execution, and maintenance. Moreover, existing approaches, methods, techniques, and tools for modelling, simulating, validating, and monitoring single digital twins must now address the increased complexity in federation scenarios. These scenarios introduce new challenges, such as digital twin identification, shared metadata, cross-digital twin communication and synchronization, and federation governance. The KDT Joint Undertaking MATISSE project tackles these challenges by aiming to provide a model-driven framework for the continuous engineering of federated digital twins. It leverages model-driven engineering techniques and practices as the core enabling technology, with traceability serving as an essential infrastructural service for the digital twins federation. In this paper, we introduce the MATISSE conceptual framework for digital twins, highlighting both the novelty of the project's concept and its technical objectives. As the project is still in its initial phase, we identify key research challenges relevant to the DATE community and propose a preliminary research roadmap. This roadmap addresses traceability and federation mechanisms, the required continuous engineering strategy, and the development of digital twin-based services for verification, validation, prediction, and monitoring. To illustrate our approach, we present two concrete scenarios that demonstrate practical applications of the MATISSE conceptual framework.**

***Index Terms*—Model-Driven Engineering, Federated Digital Twins, Continuous System Engineering, Verification & Validation**

I. Introduction

Digital Twins (DTs) are virtual representations of real-world entities which enable organizations to predict, monitor, verify, and test systems using real-time data [1]. Despite their advantages, DTs integrate complex technologies posing challenges across the systems engineering life-cycle. These challenges notably include managing heterogeneous artifacts, ensuring interoperability [2], supporting V&V, achieving standardization [3], and maintaining continuous development. These issues become more pronounced in the context of federated DTs. According to Gartner, 61% of companies using DTs employ at least two in isolation, indicating that "true integration remains complex and requires advanced integration and information management skills"[1]. In parallel, the EU also recognizes the potential of DTs and promotes their widespread adoption. The ECS-SRIA[2] is a reference document for EU programmes, outlining priorities to achieve strategic advantages for Europe. It identifies DTs as a key area of the digital industry. Among the main topics are: (i) model-driven design and DTs to support the continuous system and software integration and development, (ii) continuous integration, verification, and validation (with and without AI) using model-driven design technologies and DTs. We believe that advancing the continuous engineering of federated DTs requires addressing the following Research Challenges (RCs):

RC1: Manage the complexity and ensure seamless integration of multiple interconnected DTs. We need to specify structured model-driven frameworks to support in practice the engineering of federated DTs.

RC2: Allow the continuous validation of DTs to maintain their accuracy and reliability during their full life cycle. We need to design and develop robust model-driven frameworks to support such a continuous validation of

[1]https://emtemp.gcom.cloud/ngw/globalassets/en/publications/documents/2023-gartner-top-strategic-technology-trends-ebook.pdf
[2]ECS-SRIA: https://ecssria.eu

978-3-9826741-0-0/DATE25/

DTs.

RC3: Support the prediction, verification, and monitoring of industrial systems via their DTs. We need to create flexible, reusable, and generic (domain-independent) DT-based services supporting such activities.

Working in this direction, the HORIZON-KDT-JU funded MATISSE project aims to deliver a novel model-driven framework for the continuous engineering of federated Digital Twins (DTs) in the context of industrial systems. This 3-year European project, launched in September 2024, involves 30 organizations across 7 countries. MATISSE focuses on developing methods and tools for efficient, continuous engineering and validation of industrial systems supported by DTs. To this end, it leverages a combination of model-driven, data-driven, and cloud techniques. The project aims to boost industrial productivity, reduce development costs, and shorten time-to-market through the adoption of DTs for complex, large-scale industrial systems. In this paper, we introduce the MATISSE conceptual framework for digital twins, highlighting both the novelty of the project's concept and its technical objectives. As an essential part of MATISSE, we employ Model-Driven Engineering (MDE) methods and techniques in order to define a framework for the efficient and continuous engineering of federated DTs. This framework aims to facilitate the creation of a shared digital twin of a system through common federation mechanisms. By leveraging this framework, we intend to develop DT-based services for verification, prediction, and monitoring. Federation will be primarily achieved through MDE by relying on standard general-purpose languages (e.g., UML/SysML), domain-specific languages (DSLs), the Functional Mockup Interface (FMI), as well as model transformations and views. Although the project is still in its initial phase, we identify key research challenges relevant to the DATE community and propose a preliminary research roadmap. Additionally, we present a research methodology and two concrete scenarios to demonstrate the practical application of this conceptual framework to be achieved within the MATISSE project.

The rest of this paper is organized as follows. Section II introduces the background of our work, as well as the related work. Section III presents four application scenarios both motivating and driving our work in MATISSE. Section IV outlines the research roadmap we plan to address in this context. Section V describes the approach and supporting framework we propose accordingly. We notably explain how we want to (i) address the previously identified challenges associated with federated DTs, and (ii) provide a potential implementation of the proposed research roadmap. Finally, Section VI concludes the paper.

II. Background and Related Work

This section presents the background and related work concerning our proposed approach.

A. Background

Model-driven engineering. MDE is an engineering paradigm that relies on models as purposeful abstractions of complex (software) systems. As machine-readable artefacts, models can be manipulated throughout the whole system's life cycle. MDE includes pillar concepts including models, metamodels, megamodels, model transformations and views [4]. Overall, it offers a body of knowledge providing support for improving automation and productivity in terms of both engineering process and quality of the engineered system [4].

Digital Twin. A DT is a virtual representation of an actual system that is continuously updated with real-time data throughout its life cycle and, at the same time, can interact with and influence the system [5]. DTs are considered the technological backbone for enhancing the understanding and management of modern and existing systems in various domains. Kritzinger et al. introduce the concept of Digital Model (DM) as a digital representation without any form of automated data exchange with the system [6]. Instead, a Digital Shadow (DS) supports a one-way automated data flow between the actual system and its virtual counterpart. Finally, a DT provides the highest level of data integration, with automatic data flow to/from the physical system.

Digital Twin Federation. Jeong et al. proposed a 5-step evolutionary model for DTs from virtualization to intelligent services [7]. DT federation is the fourth evolutionary step, building atop evolutionary steps requiring synchronization, validation, and simulation capabilities. Vergara et al. [8] defined Federated Digital Twin (FDT) as an interconnection among autonomous DTs in the virtual space and presents four architectural styles. Through federation, optimized single DTs cooperate with each other in order to build up large-scale DTs (e.g., smart cities, energy grids, train systems, or other critical infrastructures). In parallel, a few approaches introduced the concept of *DT of a System or System-of-Systems*, representing a structured network of integrated DTs to cope with the heterogeneity of underlying systems [9], [10].

Traceability. Traceability refers to the capability of representing, understanding, and analyzing relationships between different kinds of software and systems artefacts. Model-driven traceability mechanisms can be crucial for supporting integration requirements, particularly for DT federation. By reyling on MDE principles and techniques, traceability between DTs, their DMs and DSs can be offered as a combination of model views, weaving and transformations [11] integrated thanks to a DM megamodel.

B. Related Work

The pervasive industrial digitization process promoted by the Industry 5.0 initiative [12] played a role in the rise of the notion of DT (e.g., real-time simulation, remote updates of physical objects, visualization of results in virtual environments). As a result, the potential application domains of MDE further expanded to complex systems such as cyber-physical systems (CPS). This notably fostered research on specification and technical implementation of general-purpose (e.g., SysML [13]) and domain-specific modeling languages with dedicated MDE workbenches [14], [15] to be used in CPS engineering platforms [16], [17]. Tallat et al. [18] survey the work toward the trends in Industry 5.0 while presenting DT categories and available frameworks. Still, despite DT being a well-established technology, there is no clear and solid agreement on the principles, roles, and features of DT technology (cf. also Section II-A). Hyre et al. tried to eliminate this confusion by defining an initial conceptual framework for constructing DTs for any systems [19]. Moreover, Zheng et al. interpreted the concept and characteristics of DT and proposed an application framework of DT for product lifecycle management [20]. Fend et al. [21] presented a language and code generation framework for the model-driven development of mobile CPS devices, a system-wide DT. Shangguan et al. proposed a hierarchical DT model framework for continuous CPS design, with an industrial robot application to demonstrate the efficacy of the model framework [22]. Finally, Tao et al. presented a new method for product design development, based on the DT approach in the CPS domain, and a related framework called DT-driven Product Design (DTPD) [23]. Recently, Lehner [24] presented an ongoing effort towards a model-driven platform for engineering holistic DTs. To the best of our knowledge, while general concepts, frameworks, and guidelines exist for using DTs in industrial settings (e.g., for CPSs), there is a lack of unified approach and framework applying across different domains and supporting the continuous engineering of federated DTs. The work introduced in this paper makes a concrete step in this direction.

III. Application scenarios

DTs are being more and more used across various sectors, from space applications to critical infrastructures within industry. This section presents several application scenarios provided by our industrial partners in MATISSE, highlighting the challenges of effectively engineering DTs. For each application scenario, we identify the underlying research challenges (RCs) and demonstrate how a framework for the efficient and continuous engineering of federated DTs could help in addressing these challenges.

Microgrids. Microgrids consist of prosumers who provide, store, and consume energy using solar panels, batteries, grid interconnections, and power outlets. When a car with a battery plugs in, the system decides where to draw energy, how much to charge, and which prosumers can pull energy from the battery. Local renewable sources can stress devices and grid architecture, with high-speed wall boxes causing unpredictable high loads due to sparse charging events. Additionally, the demand from e-charging stations often misaligns with local renewable supply, thus necessitating energy storage. DTs can manage microgrids and support the optimization of their operation for e-vehicles. Our framework can help structure and integrate DTs of different energy sources, storage systems, and consumption patterns (RC1). It can also ensure that the DT remains an accurate representation of the real-world microgrid throughout its lifecycle (RC2). Finally, it can facilitate the definition of services for optimizing energy distribution, predicting demand and supply mismatches, and monitoring grid component health (RC3).

Renewable energy communities. Renewable Energy Communities (RECs) promote local generation by prosumers to increase renewable energy adoption. Efficient RECs connect many consumers/prosumers, thus maximizing local production and consumption matches. RECs address logistical, financial, technical, regulatory, and market barriers that discourage residential renewable adoption. Managing RECs involves balancing user requests, energy availability, prices, and grid needs. This notably requires a complex network of peer-to-peer energy sales, load-shifting, energy storage, and consumption optimization supported by sensors, meters, and actuators. Many users lack the space, capital, or expertise for renewable assets. RECs achieve a critical mass by offering economies of scale, professional advice, and trade opportunities, by reducing costs and adding value. However, this increases technological and economic complexity which must be managed by an automated system. DTs of energy communities can help to study interactions and identify optimization opportunities. Our framework can help to structure and manage the integration of multiple interconnected DTs, in order to efficiently connect consumers/prosumers and optimize the local production and consumption requirements (RC1). Continuous validation ensures the DT remains accurate as user requests and energy availability fluctuate (RC2). Additionally, DT-based services can optimize energy distribution, predict demand and supply mismatches, and monitor network performance to ensure efficient and reliable REC operation (RC3).

Train control and management system. The electric traction systems and the Train Control and Management System (TCMS) are two key systems in modern trains. The traction system uses advanced sensors and control equipment in order to manage performance, energy efficiency, and safety. TCMS, built on open standard IP technology, integrates all control and communication functions onboard. It acts as the central hub of the distributed control system. Both systems are adaptive, configurable, and safety-critical. In this context, increasing testing efficiency early in development is a significant challenge. Creating, selecting, and reusing test cases from specifications and models is difficult and can miss faults. To overcome this, DTs can identify requirement-based test cases, invalid states, and conflicting requirements. Our framework may improve test automation and the reuse of test scenarios (RC2). Testing in projects is often expensive and manual. It also involves creating and executing test cases, generating reports, and analyzing bugs leading to slow feedback loops. Applying DTs aims to find defects early in the development process, reducing costs and delivery times.

Critical infrastructure. In Sweden, there are 23,000 bridges, with 10% older than 100 years and 50% older than 50 years. Maintenance and repair costs are constantly increasing, and frequent safety evaluations are necessary. Bridge owners need new tools and methods to handle the complexity of monitoring and operating these bridges. The Swedish Transport Administration wants to monitor bridges for early detection of potential damages like cracks or material fatigue, and to evaluate the effects of installing dampers. Bridges are equipped with various sensors, including strain gauges, accelerometers, displacement, bridge-weigh-in-motion, and temperature sensors. ML algorithms analyze sensor signals from each train passage to calculate material damages and detect anomalies. This data is then transferred to a cloud database. A DT can significantly enhance the understanding of sensor signals and their impact on known damages. Our framework aims at enabling seamless integration and effective use of sensor data to predict and understand structural behaviours and damages (RC1). It will also ensure that the DT accurately represents real-world bridge conditions, thus facilitating the reliable monitoring and early detection of potential damages throughout the bridges' life cycle (RC2). Finally, flexible DT-based services will improve the ability to monitor bridge conditions, train new engineers, and refine maintenance and repair strategies (RC3). This way, the safety and longevity of the infrastructure can be ensured.

Potential of Federated DTs. All these scenarios highlight the necessity of adopting federated DTs for industrial systems, notably for managing and addressing emergent behaviours. For instance, integrating DTs in the context of microgrids and RECs can provide a more comprehensive view on energy production, storage, and consumption. In the general case, federated DTs enable a better data sharing and integration. This notably facilitates the work of analytics and ML models to predict and mitigate issues before they arise. This also allows for precise optimization and real-time management, thus enhancing efficiency and reliability (e.g., of both microgrids and RECs). Moreover, it allows enhancing the infrastructure's stability and sustainability, possibly supporting more resilient and adaptable ecosystems. Similarly, federating DTs can provide a more holistic view on system and infrastructure health (e.g., for train systems and bridge monitoring). This improves their global stability and safety, fostering once again more resilient and adaptable ecosystems.

IV. Methodology

Based on the discussed challenges, our research methodology focuses on two main objectives: 1) achieving efficient continuous engineering of DTs (RC1 and RC2), and 2) developing services for verification and validation (V&V), and smart prediction of system qualities (RC3).

A. Efficient Continuous Engineering and V&V of DTs

We begin by reviewing state-of-the-art MDE techniques to select those suitable for engineering DTs. DTs will be realized by combining and validating design-time artefacts (DMs) and runtime data/models (DSs) [6]. Moreover, DTs will be updated via a feedback loop in a model-driven continuous engineering process, by leveraging traceability and federation mechanisms.

Model-Based Structure and Communication Capabilities. The first gap involves designing a conceptual framework and its practical realization to enable continuous engineering and V&V of DTs. This model-driven framework must include horizontal communication APIs/interfaces between DMs, and vertical ones between DMs and DSs. These interfaces will support the continuous engineering of DTs, establishing a model-driven feedback loop between DMs and DSs.

Model-Driven Mechanisms for Traceability and Federation. The second gap involves designing model-driven traceability and federation mechanisms for DTs, which can then be integrated into the previously mentioned conceptual framework. Traceability should support engineering activities through horizontal (between DMs) and vertical (between DMs and DSs) mechanisms. Federation must allow DTs to be combined into structured networks of interconnected DTs.

Model-Driven Support for the Continuous V&V of DTs. The third gap involves designing and developing model-driven support for a continuous validation strategy for DTs. This should build on the proposed conceptual framework and leverage the associated traceability and federation mechanisms. The approach will rely on open and/or international standards used in the validation solutions, notably based on the application scenarios presented in Section III.

B. Services for V&V and Smart Prediction of System Qualities

We aim to develop validation services using DTs to assess systems based on various quality attributes relevant to the identified scenarios. These services will enable early fault detection and correction, as well as smart prediction of system quality through data analysis, by leveraging AI/ML techniques.

Smart Test Generation Methods from Digital Twins. The first gap involves establishing methods for selecting and generating test cases using DTs. Since the space of possible test scenarios is too vast to be tested on hardware, more effective strategies using DTs need to be developed. This will include defining quality criteria and metrics for functional and non-functional testing. Based on these, methods for selecting quality test scenarios from DT models will be created. Moreover, methods will be developed to generate test inputs and outputs using techniques like model-driven testing, search-based testing, and combinatorial testing. Finally, the adaptation of test scenarios based on feedback from results and system or DT model changes will also be addressed.

Smart Test Result Evaluation, Fault Localization, and Automated Feedback. The second gap involves the evaluation and analysis of test results and DT execution data to provide automated rapid feedback. This will require developing methods for comparing test and execution data from the DTs and system, using techniques like anomaly detection, fault injection or differential testing. Moreover, methods for identifying and localizing faults and errors in the system or DT models will be created. Automated feedback mechanisms for developers and testers will also be developed using natural language generation and recommendation systems. Finally, feedback will be integrated into the DT engineering and V&V process through model-driven techniques such as model transformation, repair, and updates.

Monitoring and Smart Quality Prediction. The third gap involves developing intelligent prediction algorithms to support quality assurance under uncertainties. This will require methods for collecting and analysing data through smart

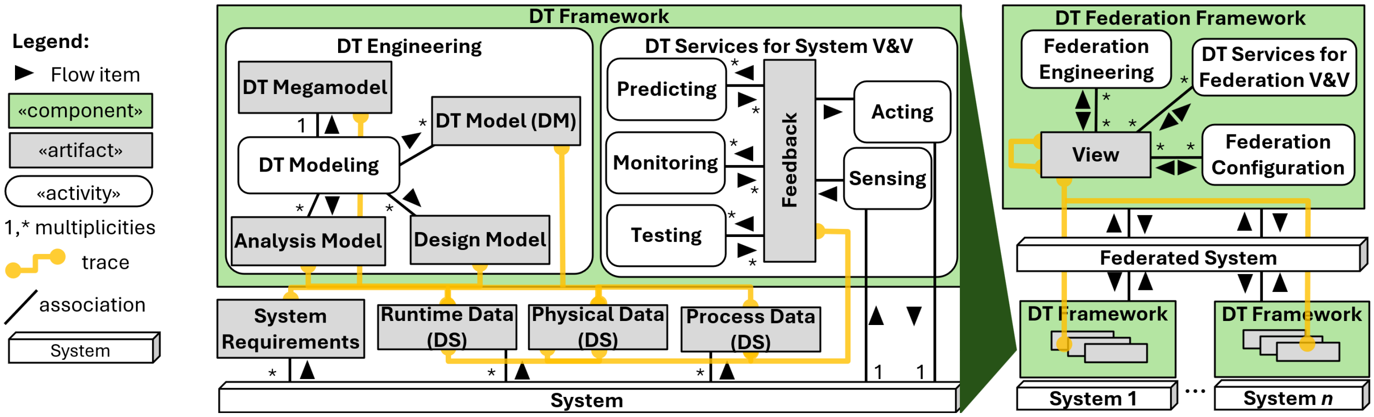

Fig. 1. Overview of the proposed conceptual framework

probes that monitor system state, performance, environment, and user interactions. Additionally, techniques like ML, data mining or statistical analysis will be used to learn from the data collected by the smart probes and DT. Methods such as regression, classification, and clustering will estimate system or DT model quality and predict the likelihood of faults and performance issues. Finally, techniques for extracting patterns and rules describing system behaviour and factors affecting quality will be developed.

V. Approach

An overview of the proposed conceptual framework is depicted in Figure 1. We first rely on a *DT Engineering* toolkit that defines a *Digital Twin (DT)* as a network of interconnected *Digital Models (DM)* and *Digital Shadows (DS)*. On the one hand, DMs can represent both software and hardware components of the *System* being engineered, or any relevant *System* information (e.g., analysis or design), depending on the needs of stakeholders such as system architects, users, and technology providers. Each DM conforms to a metamodel that specifies the DT modeling concepts and their relationships. On the other hand, DSs are *Runtime, Physical or Process Data Sets* that are continuously collected, processed (e.g., labelled, filtered), and updated. These DSs, along with the *System Requirements*, serve as main inputs for the framework, thus forming a *Feedback* traceability loop with the corresponding DMs. We propose a continuous DT engineering process, based on the interaction between DMs and DSs, by leveraging MDE principles and techniques such as model-to-model transformation [25], model weaving [26], model views [11], and megamodeling [27]. To realize a DT, data and models from various engineering disciplines must be integrated, synchronized, and managed. As described in Section IV, these artefacts can exist in various formats, standardized or not, and should be shared and reused across the solutions contributing to the framework.

In the proposed framework, we also provide *DT Services for Systems Verification & Validation (V&V)* alongside the *DT Engineering* toolkit. These services leverage the DT, particularly its heterogeneous DMs, for 1) *Predicting* system behaviours and their potential evolution, 2) *Monitoring* the system and its (actual or simulated) execution in various contexts, and 3) *Testing* system characteristics and their validity in different situations. Depending on the activity, different types of DMs (e.g., mathematical, machine learning, data-based, software, or engineering models) can be selected. To scale up for large and complex systems, we introduce the concept of *Federation of Twins*, shown on the right-hand side of Figure 1. Engineering and V&V services for individual DTs must evolve to support federated engineering practices and V&V services, including configuration management for federations (e.g., identification and governance). Unlike monolithic DTs, this framework uses model-driven principles and techniques to ensure DT federation. DT federation will be achieved through an overarching *DT megamodel* [27] that represents all the involved DMs, DSs, and their interrelations. From this *DT megamodel*, one or more specific *Views* can be built to support the targeted engineering activities. Communication and synchronization between DTs will occur via these views and the megamodel, thus facilitating regular exchanges and updates between DMs and DSs. Moreover, DTs contain various information including product dimensions, configuration settings (e.g., firmware), and design data, all stored in corresponding models. While DT data may be collected from multiple sources and stored in a distributed manner, DT models and artifacts require interoperability and traceability for reuse. In a federation context, this involves multiple DTs communicating with one another, sharing modules, or containing other DTs. In order to efficiently store and retrieve all the required modelling artifacts, we plan to rely on state-of-the-art model repositories and complementary solutions.

To summarize, DT engineering must be supported by innovative model-driven techniques that efficiently describe, handle, and manage diverse heterogeneous artifacts (e.g., design models, metamodels, transformations, logs) and their industrial environments [28]. This also involves managing the associated relationships between all these artifacts (e.g., traceability, refinement, combination). To this end, we will notably rely on megamodels and model views to track such relationships at various levels. The ultimate goal is to foster integration and collaboration over related models and artifacts.

VI. Conclusion and Outlook

This paper presented a conceptual framework, along with corresponding challenges and methodology, for the continuous engineering of federated digital twins, as part of the HORIZON-KDT-Ju MATISSE project. This project aims to design, develop, and test a framework that integrates methods and tools for the efficient continuous engineering and V&V of industrial systems through DTs. The framework will leverage model-driven, data-driven, and cloud techniques in order to provide validation and verification services. This framework is expected to significantly enhance the engineers productivity and the overall quality of the target industrial systems.

Acknowledgment

This research work has been funded by the Key Digital Technologies (KDT) Joint Undertaking through the project MATISSE, grant agreement No. 101140216.

References

[1] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, and A. Calinescu, "Digital twins: State of the art theory and practice, challenges, and open research questions," *Journal of Industrial Information Integration*, vol. 30, p. 100383, 2022.

[2] E. Ferko, A. Bucaioni, P. Pelliccione, and M. Behnam, "Analysing interoperability in digital twin software architectures for manufacturing," in *European Conference on Software Architecture*. Springer, 2023, pp. 170–188.

[3] ——, "Standardisation in digital twin architectures in manufacturing," in *2023 IEEE 20th International Conference on Software Architecture (ICSA)*, 2023, pp. 70–81.

[4] S. Kent, "Model driven engineering," in *International conference on integrated formal methods*. Springer, 2002, pp. 286–298.

[5] F. Bordeleau, B. Combemale, R. Eramo, M. Van Den Brand, and M. Wimmer, "Towards model-driven digital twin engineering: Current opportunities and future challenges," in *Systems Modelling and Management: First International Conference, ICSMM 2020, Bergen, Norway, June 25–26, 2020, Proceedings 1*. Springer, 2020, pp. 43–54.

[6] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline*, vol. 51, no. 11, pp. 1016–1022, 2018.

[7] D.-Y. Jeong, M.-S. Baek, T.-B. Lim, Y.-W. Kim, S.-H. Kim, Y.-T. Lee, W.-S. Jung, and I.-B. Lee, "Digital twin: Technology evolution stages and implementation layers with technology elements," *IEEE Access*, vol. 10, pp. 52 609–52 620, 2022.

[8] C. Vergara, R. Bahsoon, G. Theodoropoulos, W. Yanez, and N. Tziritas, "Federated digital twin," in *IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2023, pp. 115–116.

[9] L.-T. Reiche, C. S. Gundlach, G. F. Mewes, and A. Fay, "The digital twin of a system: A structure for networks of digital twins," in *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–8.

[10] M. Tisi, H. Bruneliere, J. de Lara, D. Di Ruscio, and D. Kolovos, "Towards Twin-Driven Engineering: Overview of the State-of-The-Art and Research Directions," in *IFIP Conference on Advances in Production Management Systems (APMS 2021)*, 2021, pp. 351–359.

[11] H. Bruneliere, E. Burger, J. Cabot, and M. Wimmer, "A feature-based survey of model view approaches," *Software & Systems Modeling*, vol. 18, pp. 1931–1952, 2019.

[12] P. K. R. Maddikunta, Q.-V. Pham, P. B, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.

[13] E. Ferko, L. Berardinelli, A. Bucaioni, M. Behnam, and M. Wimmer, "Towards interoperable digital twins: Integrating sysml into aas with higher-order transformations," in *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 2024, pp. 342–349.

[14] B. Combemale, J. DeAntoni, B. Baudry, R. B. France, J.-M. Jézéquel, and J. Gray, "Globalizing modeling languages," *Computer*, vol. 47, no. 6, p. 68–71, Jun. 2014. [Online]. Available: http://dx.doi.org/10.1109/MC.2014.147

[15] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework*. Pearson Education, 2008.

[16] H. Thompson, M. Reimann, D. Ramos-Hernandez, S. Bageritz, A. Brunet, C. Robinson, B. Sautter, J. Linzbach, H. Pfeifer, V. Aravantinos *et al.*, *Platforms4CPS, Key Outcomes and Recommendations*. Steinbeis-Edition, 2018.

[17] A. Sadovykh, A. Bagnato, I. Quadri, A. E.-D. Mady, L. D. Couto, S. Basagiannis, and M. Hasanagic, "SysML as a Common Integration Platform for Co-Simulations: Example of a Cyber Physical System Design Methodology in Green Heating Ventilation and Air Conditioning Systems," in *Proceedings of the 12th Central and Eastern European Software Engineering Conference*, ser. CEE-SECR'16, 2016, pp. 1–5.

[18] R. Tallat, A. Hawbani, X. Wang, A. Al-Dubai, L. Zhao, Z. Liu, G. Min, A. Y. Zomaya, and S. Hamood Alsamhi, "Navigating industry 5.0: A survey of key enabling technologies, trends, challenges, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 1080–1126, 2024.

[19] A. Hyre, G. Harris, J. Osho, M. Pantelidakis, K. Mykoniatis, and J. Liu, "Digital twins: Representation, replication, reality, and relational (4rs)," *Manufacturing Letters*, vol. 31, p. 20–23, Jan. 2022.

[20] Y. Zheng, S. Yang, and H. Cheng, "An application framework of digital twin and its case study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 3, p. 1141–1153, 2018.

[21] A. Fend and D. Bork, "Cpsaml: a language and code generation framework for digital twin based monitoring of mobile cyber-physical systems," in *Companion Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*. ACM, 2022, p. 649–658.

[22] D. Shangguan, L. Chen, and J. Ding, "A hierarchical digital twin model framework for dynamic cyber-physical system design," in *Proceedings of the 5th International Conference on Mechatronics and Robotics Engineering*, ser. ICMRE'19, 2019, p. 123–129.

[23] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Y. C. Nee, "Digital twin-driven product design framework," *International Journal of Production Research*, vol. 57, no. 12, p. 3935–3953, Feb. 2018.

[24] D. Lehner, "A model-driven platform for engineering holistic digital twins," in *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2023, pp. 179–185.

[25] F. Jouault and I. Kurtev, "On the interoperability of model-to-model transformation languages," *Science of Computer Programming*, vol. 68, no. 3, pp. 114–137, 2007.

[26] J.-M. Jézéquel, "Model driven design and aspect weaving," *Software & Systems Modeling*, vol. 7, pp. 209–218, 2008.

[27] J. Bézivin, F. Jouault, and P. Valduriez, "On the need for megamodels," in *19th Annual ACM conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'04)*, 2004, pp. 1–9.

[28] A. Bucaioni, A. Di Salle, L. Iovino, I. Malavolta, and P. Pelliccione, "Reference architectures modelling and compliance checking," *Software and Systems Modeling*, vol. 22, no. 3, pp. 891–917, 2023.