A Model-Driven Digital Twin for Manufacturing Process Adaptation

Patrick Spaney
Steffen Becker
University of Stuttgart, ISTE
Stuttgart, Germany
www.iste.uni-stuttgart.de

Robin Ströbel
Jürgen Fleischer
Karlsruhe Institute of Technology, wbk
Karlsruhe, Germany
www.wbk.kit.edu

Soraya Zenhari
Hans-Christian Möhring
University of Stuttgart, IfW
Stuttgart, Germany
www.ifw.uni-stuttgart.de

Ann-Kathrin Splettstößer
Andreas Wortmann
University of Stuttgart, ISW
Stuttgart, Germany
www.isw.uni-stuttgart.de

***Abstract*—Digital twins are a means to better understand, engineer, and use cyber-physical systems. In manufacturing, digital twins can optimize production, prevent failures, and save resources. To consolidate the different approaches to digital twins in manufacturing, ISO 23247 defines the essential functional elements of a digital twin. We present a model-driven digital twin exemplar that realizes part of this standard to analyze milling processes. Our digital twin reference architecture includes a digital twin service component that manages and connects different services to adapt the manufacturing process according to a given objective. Two digital twin services, for the adaptation of component and tool conditions and for geometric error adaptation, illustrate the potential of this reference architecture. The digital twin connects to an industrial milling machine via domain-specific languages. This exemplar uses models at design time and at runtime to separate the concerns of software engineers and domain experts and leverages these models to understand and optimize the use of industrial machine tools.**

***Index Terms*—Digital Twin, Model-Driven Development, Software Architecture, Process Adaptation**

I. Introduction

Research and industry employ *Digital Twins (DTs)* [1], [2] to monitor and control *Cyber-Physical Systems (CPSs)* in various domains, including automotive [3], biology [4], medicine [5], manufacturing [6], and many more. They promise vast potential to reduce development costs and time, optimize operations, and improve our understanding of the represented systems. In manufacturing, DTs often are twinning *Cyber-Physical Production Systems (CPPSs)* to improve machining operations [7]. And while there are many publications on DTs in manufacturing, few actually describe the architecture of the DTs being used [7]. This prevents understanding DTs in manufacturing, hampers analyzing their interoperability, and complicates their widespread adoption.

We present an exemplar of a model-driven DT for manufacturing in the sense of [1], *i.e.,* for us, a DT is a software system using data, models, and services to purposefully represent and manipulate its original CPS. This exemplar uses models at the design time of the DT to specify its structure and overall behavior, which need to be created by software experts. During instantiation and at runtime, the DT uses models to specify its application-specific and CPS-specific behavior, *i.e.,* models that need to be created and maintained by domain experts.

In the following, Sec. II highlights selected foundations of this exemplar before Sec. III describes its component-based architecture. Afterward, Sec. IV discusses our model-driven approach to connecting the DT with the twinned CPS, and Sec. V describes its manufacturing services. Sec. VI concludes.

II. Background

The capability of machining processes represents the basis for the quality of the produced parts. Despite coating and material technologies having been advanced for decades to extend tool life, tool wear is an inevitable phenomenon that can significantly impact the quality of the workpiece [8]. Therefore, the regular monitoring of tools would be highly advantageous in evaluating the overall process control in modern manufacturing systems [9]. There are many factors involved in tool wear, such as abrasion, adhesion, diffusion, and chemical interactions, which contribute to the process of tool wear [10]. The condition of the machine tool and its components represent another important aspect of quality. Therefore, several approaches for the determination of the component states, such as the motor as a relevant part of a feed axis, were developed in the past [11]. The focus is usually on mapping the identified properties representing the component condition to form the basis for, e.g., condition monitoring or predictive maintenance. However, a benefit only arises when this information is used to adapt or optimize the process and thus to be able to meet the required component quality. Therefore, the integration into easily applicable DTs represents

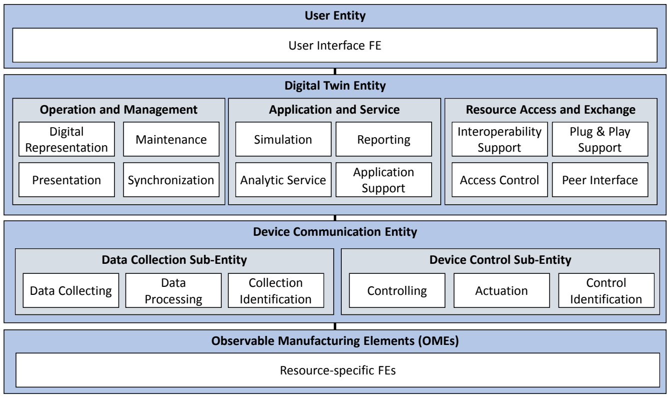

Fig. 1: Conceptual digital twin model of ISO 23247 [14]

an important basis for the generation of added value, as these enable reacting to change in the condition and adapting the machining processes.

We employ the MontiArc [12] architecture description language to model the software architecture of our DT exemplar. MontiArc is a textual component & connector architecture description language with semantics based on the focus calculus [13]. In MontiArc, architectures are hierarchical compositions of components that exchange messages over typed and directed channels between them. Components are either atomic and contain behavior specifications themselves or composed of other components. MontiArc models can be translated into target languages, including C++, Java, Mona, Python, and more.

ISO 23247 [14] conceptually describes a DT framework for manufacturing (*cf.* Fig. 1). This framework comprises three layers of *Functional Entities (FEs)* that provide functionality over *Observable Manufacturing Elements (OMEs)*, where OMEs are assets (physical or not) that can be observed (*e.g.,* a manufacturing plant, a process, …). The device communication layer contains FEs to gather and process data from OMEs and to control them. The digital twin layer uses the device communication FEs to represent, manage, simulate, and maintain the OMEs. The user interface entity layer provides user interface FEs based on the other two layers.

The systematic engineering of software systems, such as DTs, requires standardized protocols and common information models to ensure connectivity and interoperability between the DTs and twinned CPSs. *Open Platform Communication Unified Architecture (OPC UA)* is a set of standards providing technical protocols and its own meta-model (roughly resembling the integration of class diagrams with object diagrams) for describing CPPS communication interfaces [15]. This makes systems equipped with OPC UA a natural target for a model-driven approach to connectivity, enabling the generation of communication endpoints from explicitly modeled OPC UA data [16]. Consequently, the model-driven approach alleviates the need for tedious and error-prone manual integration of DTs with their corresponding CPPS. However, previous approaches require the DT to be modeled in accordance with the existing CPPS's information model. To extend model-driven CPPS connectivity to settings with heterogeneous information and thus encourage the reuse of software components, we propose a family of domain-specific languages (DSLs) that support mappings between the CPPS's OPC UA address space and the interfaces of the software components consuming the data.

III. Model-Driven Digital Twin Architecture

We employ a model-driven approach to engineering DTs implementing ISO 23247 that relies on (1) class diagrams to describe the domain data types the DT operates with and its Digital Shadows (DSs) [17], (2) a MontiArc [12] software architecture (structure and behavior) of the DT itself, (3) code generators translating the class diagrams into an enterprise information system [18] on top of an SQL database, and (4) code generators translating the DT's MontiArc architecture into an implementation using the same database [19]. Hence, from the input models, we obtain a DT implementation that includes a web-based dashboard [19]. After code generation, the DT needs to be configured with application-specific behavior models that it interprets at runtime. These include event-condition-action models and case-based reasoning rules and must be specified by domain experts familiar with the twinned CPS [19].

The architecture of this exemplar (*cf.* Fig. 2) extends an existing DT architecture [19] that realizes a MAPE-K loop over an OPC UA [20] connection to the twinned CPS and a data lake that may comprise further information about that CPS. This data lake also contains the database that the generated enterprise information system is connected to. The `Monitor` component investigates the data lake and produces DSs [17] that are analyzed by the `Evaluator` using the application-specific event-condition-action rules and a shared `KnowledgeBase`. In case the `Evaluator` identifies problems, it emits a reasoning goal to the `Planner`, which uses an application-specific case base to find a solution, which is sent to the `Executor`, who translates it to OPC UA commands and sends these to the CPS. Both `Monitor` and `Executor` implement the device communication layer of ISO 23247. The components `Evaluator`, `KnowledgeBase`, and `Planner` implement FEs of the ISO 23247 digital twin layer. All of these components are connected to the dashboard [19] (realizing part of the ISO 23247 user interface layer).

The presented exemplar introduces novel digital twin services for sustainability in compliance with ISO 23247 that investigate milling processes with regard to tool deterioration and enable action before the milling tool quality affects the intended processes negatively. This, *e.g.,* prevents suboptimal products, extensive energy use with bland tools, and waste. To this end, the `DigitalTwinServices` component yields subcomponents for individual services, which connect to the `ServiceManager`, who sends queries (such as registration of new kinds of DSs) to the `DigitalTwinEngine`. The services also realize FEs on the ISO 23247 digital twin layer.

The following sections detail the connectivity and the services of this architecture.

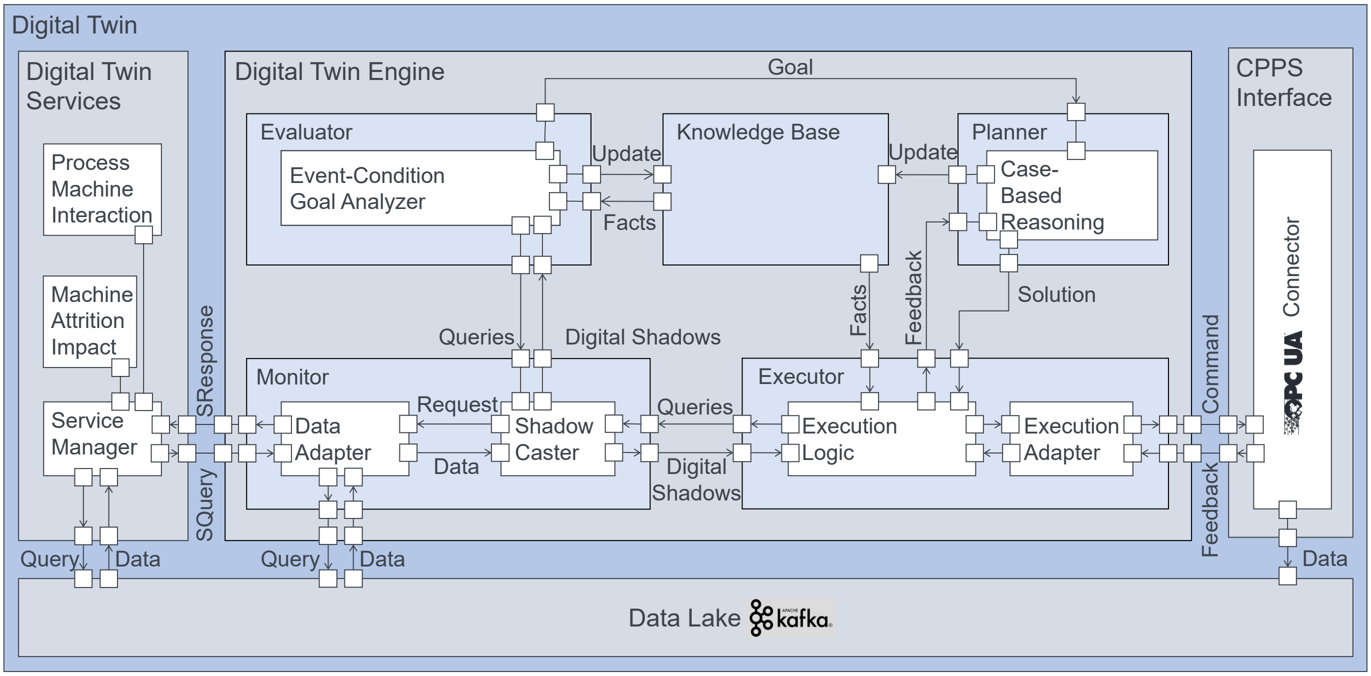

Fig. 2: Model-driven, component-based digital twin architecture implementing ISO 23247

IV. Model-Driven Connectivity

We extend our model-driven approach to the DT's connectivity, integrating internal components via generated messaging endpoints [21] and reading the CPPS data from its OPC UA server via a generated client component. The messaging endpoints connect to a middleware solution, for which we chose Apache Kafka [22] due to its capabilities regarding high-throughput data streaming and storage, making it a suitable choice to decouple the overall architecture from the concrete realization of the DT services. To provide connectivity in the presence of heterogeneous information models within the context of CPPSs, we use a set of external DSLs to describe software components and CPPS interfaces, including engineering units and sampling rates, and to describe simple mappings between these interfaces. These interface descriptions serve as input to generate the Kafka Endpoints and, in the OPC UA case, as a better readable representation of the existing model, while the mapping is required to generate transformation code, bridging the gap between the CPPS's and the DT's interfaces.

We illustrate the integration of a `DigitalTwinService` on the machine tool condition service in Listing 1. The CPPS provides its sensor and control data as variables that are addressable via unique node IDs within the exposed `address_space` of the CPPS's OPC UA server (top listing). We model service `interfaces` in a similar fashion, but we omit the OPC UA-specific addressing and additionally differentiate between input and output parameters of the service (middle listing). For both `address_spaces` and `interfaces`, the respective attributes must specify a type and may optionally declare a unit. Furthermore, any component interface declares a sampling rate for its incoming pa-

```
address_space MillingMachine {
  // Attribute | Type | Unit | OPC UA node
  tool_pos_x Float cm @ 1:MiMa_Pos
  tool_pos_y Float cm @ 1:MiMa_Pos
  tool_curr Float mA @ 1:MiMa_Curr1
}
```

```
interface TConditionService {
  samplingrate 500Hz
  // Direction | Attribute | Type | Unit
  in position_x Float cm
  in position_y Float cm
  in current Float A
  out wearIndex Float[][]
}
```

```
mapping MillingMachine to TConditionService {
  tool_pos_x -> position_x
  tool_pos_y -> position_y
  tool_curr -> current
}
```

Listing 1: Excerpts of the design-time models used to describe (1) the milling machine's OPC UA address space, (2) the machine attrition impact service interface, and (3) the explicit mapping to match both interfaces

rameters. While attribute-level sampling rates are conceivable, our exemplar services do not require this and instead expect all their inputs at each update. Each interface definition is sufficient to generate the corresponding Kafka endpoint or OPC UA client code. However, for the CPPS's data to match the DT's internal format, the OPC UA connector must execute the transformation defined by the `mapping` model (bottom listing). We designed a lightweight syntax using arrows as a

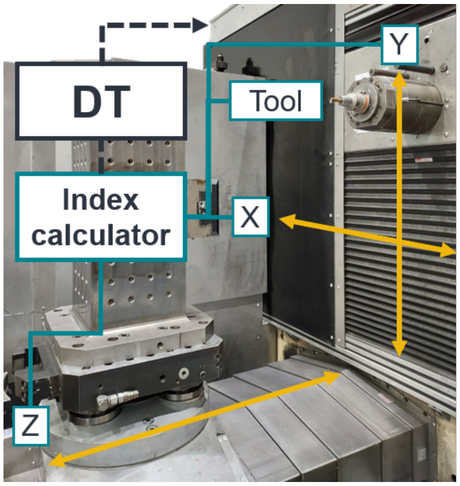

Fig. 3: Machine setup for adaptive condition mapping

mapping operator to map the `address_space` variables to the corresponding `interface` input parameters. In addition to implementing the explicitly specified matching of OPC UA variables to service parameters, the generator performs compatibility checks and infers implicit unit or data type conversions.

V. Digital Twin Services

The DT can adapt to changes in CPPS properties. This facilitates the mapping of original machine capabilities and, consequently, allows for adaptations to the actual manufacturing process. This section presents the application of the DT in mapping tool and machine conditions for a 4-axis milling machine, as well as addressing changes in geometric machine properties in a 5-axis machining center.

A. Service for Component and Tool Condition Adaptation

The condition of tools and components represents one of the most relevant factors of a machine tool. The state and the associated capabilities for the execution of various processes are subject to change over time, which is why an active adaptation of its DT is required as part of the CPPS. For the application of the adaptive DT, the 4-axis milling machine DMC 60H from 1997 retrofitted with state-of-the-art control components is available.

Fig. 3 illustrates the concept of the presented application on this machine. In order to provide the DT with continuous information about the current capability, capability indices are introduced. These indices represent the condition of axes and tools based on recurring process segments, allowing an adjustment of a given process based on the current machine capability. The index calculator generates the indices close to the machine and transfers them to the DT with minimal transmission effort if needed. This is intended to ensure the processing of relevant information in the machine environment, the DT only receives the information relevant to its assigned task.

The index calculator computes on a machine near edge device. For this, it receives access to all available machine signals by the automated machine connection presented in Sec. IV. Within one evaluation step, the system can encounter long time series, which it segments and assigns to a manufacturing process that occurs multiple times. These classified signals represent the basis for the detection of anomalies. By quantifying the deviation, it is possible to draw conclusions about the progress of the wear-based condition. The index calculator determines these deviations for all detected process segments, including the tool used and the different components of the machine (e.g., X-, Y-, Z-axis), and stores them as indices as shown in Fig. 3. All indices are structured as a matrix and made accessible for transferring the current state and associated capability. The transfer of this information to the DT can now be initiated either on change or on demand.

The transmitted indices provide the DT with knowledge of the condition present for executing a specific process at any given time. Based on this information, the DT can adjust the production process. Hence, the DT can initiate a change as soon as a tool reaches a state where it can no longer meet the quality requirements. If a poor component condition occurs frequently within a process, the DT could move the process to the other axes. Additionally, it may recommend inspecting the components.

B. Service for Geometrical Error Adaptation

A multitude of factors, including geometric, thermal, and gravity-induced errors, significantly affect the accuracy of machine tools. These inherent errors contribute to a notable discrepancy between the ideal tool path generated by the computer numerical control machine and the followed path during machining operations. The attainment of optimal performance in a machine tool crucially relies on its ability to precisely position the tool center point in relation to the workpiece dimensions. Sophisticated error compensation methods, grounded in the machine's kinematics, mitigate the impact of these errors. This process involves preemptively correcting the numerical control instructions before initiating the machining process. Subsequently, this approach integrates capability indices into the DT as performance indicators. These indices represent error compensation, ensuring the optimal performance of the machine tool.

The DT detects and rectifies deviations in the machining process by employing its dynamic and responsive capabilities. As an example consider a situation where a material inconsistency increases the resistance during milling. The DT detects this abnormality through data analysis and sends commands to adjust the milling parameters, ensuring that the material is machined accurately and meets the required specifications.

We validate the proposed method by applying it to a Grob G552 5-axis machining center to evaluate its effectiveness in enhancing the accuracy and precision of machining operations. This five-axis machine tool consists of three prismatic joints (P) and two rotary joints (R). In this study, we consider the RRPPP arrangement, as shown in Fig. 4. We use the

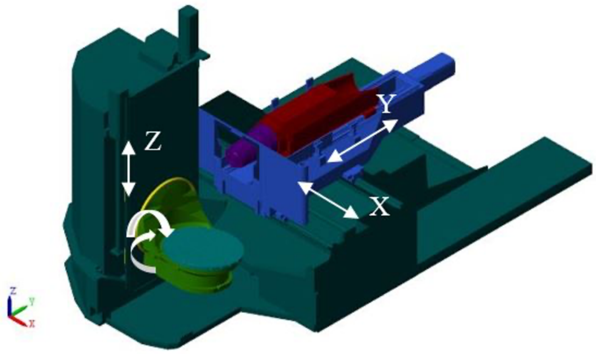

Fig. 4: 5-axis milling machine tool

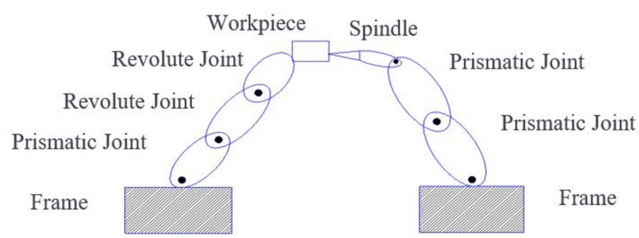

Fig. 5: Kinematic chain diagram of 5-axis milling machine tool

machine's kinematics (*cf.* Fig. 5) for error compensation and the construction of geometric error models.

VI. Conclusion

Through our exemplar of a model-driven digital twin for manufacturing process analysis, we present a real-world application of a manufacturing DT compliant with ISO 23247. Our main goal is to understand and improve the milling process carried out by the CPPS. The separation of models at design time and models at runtime enables us to leverage the knowledge of software engineers and manufacturing domain experts alike. This allows for the optimization of the manufacturing process towards more efficiency and higher product quality.

Our DT is customized to a specific CPPS. However, the architectural DT framework and connectivity approach are not bound to this application and can be reused for other implementations. Thus, our approach generally simplifies the development of manufacturing DTs.

Acknowledgment

The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the Innovations-Campus Future Mobility (ICM).

References

[1] 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.

[2] R. Eramo, F. Bordeleau, B. Combemale, M. van Den Brand, M. Wimmer, and A. Wortmann, "Conceptualizing digital twins," *IEEE Software*, 2021.

[3] X. Chen, E. Kang, S. Shiraishi, V. M. Preciado, and Z. Jiang, "Digital Behavioral Twins for Safe Connected Cars," in *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, 2018.

[4] M. Joordens and M. Jamshidi, "On The Development of Robot Fish Swarms in Virtual Reality with Digital Twins," in *13th Annual Conference on System of Systems Engineering (SoSE)*, 2018.

[5] N. Lauzeral, D. Borzacchiello, M. Kugler, D. George, Y. Rémond, A. Hostettler, and F. Chinesta, "A model order reduction approach to create patient-specific mechanical models of human liver in computational medicine applications," *Computer methods and programs in biomedicine*, vol. 170, pp. 95–106, 2019.

[6] J. Um, J. Popper, and M. Ruskowski, "Modular augmented reality platform for smart operator in production environment," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018.

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

[8] N. L. de Lacalle and A. L. Mentxaka, *Machine tools for high performance machining*. Springer Science & Business Media, 2008.

[9] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld, "Advanced monitoring of machining operations," *CIRP Annals*, vol. 59, no. 2, pp. 717–739, 2010.

[10] J. T. Black, R. A. Kohser, and E. P. DeGarmo, *Materials and processes in manufacturing*. Wiley, 2003.

[11] Y. Han and Y. Song, "Condition monitoring techniques for electrical equipment-a literature survey," *IEEE Transactions on Power Delivery*, vol. 18, no. 1, pp. 4–13, 2003.

[12] A. Butting, A. Haber, L. Hermerschmidt, O. Kautz, B. Rumpe, and A. Wortmann, "Systematic language extension mechanisms for the MontiArc architecture description language," in *Modelling Foundations and Applications: 13th European Conference, ECMFA 2017, Held as Part of STAF 2017, Marburg, Germany, July 19-20, 2017, Proceedings 13*, pp. 53–70, Springer, 2017.

[13] M. Broy and K. Stølen, *Specification and development of interactive systems: focus on streams, interfaces, and refinement*. Springer Science & Business Media, 2012.

[14] ISO, "ISO 23247–1: Automation Systems and Integration - Digital Twin Framework for Manufacturing - Part 1: Overview and general principles," tech. rep., International Organization for Standardization, 2021.

[15] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Berlin, Heidelberg: Springer, 2009.

[16] P. Bibow, M. Dalibor, C. Hopmann, B. Mainz, B. Rumpe, D. Schmalzing, M. Schmitz, and A. Wortmann, "Model-Driven Development of a Digital Twin for Injection Molding," in *International Conference on Advanced Information Systems Engineering (CAiSE'20)* (S. Dustdar, E. Yu, C. Salinesi, D. Rieu, and V. Pant, eds.), vol. 12127 of *Lecture Notes in Computer Science*, pp. 85–100, Springer International Publishing, June 2020.

[17] S. Braun, M. Dalibor, N. Jansen, M. Jarke, I. Koren, C. Quix, B. Rumpe, M. Wimmer, and A. Wortmann, *Engineering Digital Twins and Digital Shadows as Key Enablers for Industry 4.0*, pp. 3–31. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023.

[18] K. Adam, L. Netz, S. Varga, J. Michael, B. Rumpe, P. Heuser, and P. Letmathe, "Model-based generation of enterprise information systems.," in *EMISA Forum: Vol. 38, No. 1*, De Gruyter, 2018.

[19] T. Bolender, G. Bürvenich, M. Dalibor, B. Rumpe, and A. Wortmann, "Self-adaptive manufacturing with digital twins," in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, SEAMS '21, (New York, NY, USA), p. 1–4, Association for Computing Machinery, 2021.

[20] S.-H. Leitner and W. Mahnke, "OPC UA–service-oriented architecture for industrial applications," *ABB Corporate Research Center*, vol. 48, no. 61-66, p. 22, 2006.

[21] G. Hohpe and B. Woolf, *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.

[22] Apache Software Foundation, "Apache Kafka." https://kafka.apache.org, 2011.