Model-Based DevOps: Foundations and Challenges

Benoit Combemale
Jean-Marc Jézéquel
Quentin Perez
Didier Vojtisek
University of Rennes 1
263 Avenue du Général Leclerc,
F-35042 Rennes, France
benoit.combemale@irisa.fr
jezequel@irisa.fr
quentin.perez@inria.fr
didier.vojtisek@inria.fr

Nico Jansen
Judith Michael
Florian Rademacher
Bernhard Rumpe
Software Engineering,
RWTH Aachen University
Ahornstraße 55,
D-52074 Aachen, Germany
jansen@se-rwth.de
michael@se-rwth.de
rademacher@se-rwth.de
rumpe@se-rwth.de

Andreas Wortmann
Jingxi Zhang
Institute for Control Engineering of
Machine Tools and Manufacturing
Units (ISW),
University of Stuttgart
Seidenstraße 36,
D-70174 Stuttgart, Germany
wortmann@isw.uni-stuttgart.de
jingxi.zhang@isw.uni-stuttgart.de

***Abstract*—Time-to-market and continuous improvement are key success indicators to deliver for Industry 4.0 Cyber-Physical Systems (CPSs). There is thus a growing interest in adapting DevOps approaches coming from software systems to CPSs. However, CPSs are made not only of software but also of physical parts that need to be monitored at runtime. In this paper, we claim that Model-Driven Engineering can facilitate DevOps for CPSs by automatically connecting a CPS design model to its runtime monitoring, in the form of a digital twin.**

***Index Terms*—DevOps, digital twins, evolution of engineering models, models at runtime, self-adaptation, model-based software engineering**

I. Introduction

DevOps is a software development approach that bridges the traditional divide between development (Dev) and operations (Ops) teams in an organization [1]. The primary goal of DevOps is to shorten development and deployment cycles between Dev and Ops based on a more collaborative relationship and agile collaboration between both factions. To this end, DevOps introduces automation into all stages of the software development lifecycle—from integration, testing, and release to deployment and infrastructure management. As a result, software development and evolution become more efficient and reliable.

DevOps practices, while beneficial for traditional software development, introduce unique challenges when applied to the engineering of Cyber-Physical Systems (CPSs) [2]. CPSs are integrations of computation, networking, and physical processes, where embedded computers and networks monitor and control physical processes with feedback loops. The complex, intertwined nature of these systems, along with their real-time and safety requirements, poses significant challenges for DevOps applications.

For us, a digital twin is a software system that leverages CPS-related models and data to represent, predict, and prescribe CPS behavior for a specific purpose [3]. A digital twin is used at Ops time and plays the counterpart of a design model in the modeling environment at Dev time.

In this paper, we claim that Model-Driven Engineering (MDE) can facilitate DevOps for CPSs. Therefore, we introduce Model-Based DevOps (MBDO) as a novel research avenue that studies the automatic connections of CPS design models to CPS runtime monitoring in the form of digital twins. We contribute (i) a first analysis of major challenges in MBDO; and (ii) a research roadmap to tackle these challenges and derive further research opportunities for the study of MBDO. We motivate and discuss both contributions w.r.t. a case study demonstrator for sustainable Industry 4.0.

The remainder of the paper is organized as follows. Sect. II presents MBDO's potential adoption and its benefits in the context of the case study demonstrator. Sect. III introduces foundations and major challenges of MBDO based on which Sect. IV formulates a corresponding research roadmap. Sect. V presents related work and Sect. VI concludes the paper.

II. Model-Based DevOps for Sustainable Industry 4.0 Factories

Industry 4.0 factories leverage advancements from networking and digitalization to facilitate flexible and optimized production processes [4], [5]. Specifically, they enable the production of customer-oriented solutions from data gathered during a product's lifecycle. We expect MBDO to significantly benefit the evolution of traditional factories towards *sustainable Industry 4.0 factories* [6] because it supports optimizing the digital twins of factories by reintegrating knowledge from Ops into Dev models—and thus from the actual production process into its conceptualization.

To investigate MBDO in the context of sustainable Industry 4.0 factories, we first devise a demonstrator of a simplified production line. The demonstrator consists of several modules from fischertechnik construction kits[1]. We accompany each module with digital twins so that we can monitor its status, and study MBDO's potential for optimizing modules' design

[1] https://www.fischertechnik.de/en

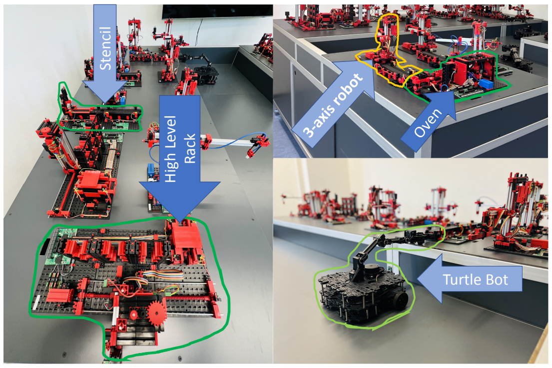

Figure 1: fischertechnik factory setup of our demonstrator. 3-axis robot, stencil machine, high level racks, oven module, and turtle bots are marked with arrows.

and behavior models. Hence, we anticipate MBDO to result not only in optimized production processes but also in valuable knowledge about the effective application of optimization techniques for a sustainable Industry 4.0 [7].

Fig. 1 shows the demonstrator and its setup for a manufacturing process of frozen yogurt by five production stations.

Each station is accompanied by a digital twin and incorporates a specialized refinement procedure, i.e., stirring, topping, and freezing, that enhances the final product. Additionally, there exist stations for logistics tasks, i.e., warehousing and sorting. The different production steps require the following machines shown in Fig. 1:

- *3-axis robots and conveyor belts:* Robots transport products between conveyor belts, and equip a grappler to lift and set products between production stations.
- *Stencil machine:* Stirring and topping of products.
- *High level racks:* Sorting and storing of raw materials and products.
- *Oven module:* Freezing of products.
- *Turtle bots:* Transportation of products between stations.

The digital twins of the demonstrator modules are compliant with ISO 23247 [8] in that they contain a digital representation of the physical production stations and are managed by a digital twin entity.

We plan to leverage the digital twins of the production stations to investigate the challenges, principles, and techniques of MBDO (Sect. III) for the optimization of production processes along the three categories of sustainability [9]:

- *Economic:* We aim to apply MBDO to simulate reconfigurations of the production process for optimization purposes. Specifically, we plan to explore multiple possible production layouts to find an optimal layout that maximizes efficiency.
- *Environmental:* MBDO-based simulations on this level provide deeper insights into resource consumption and emissions of factories that can be balanced with product quality and process efficiency w.r.t. the economic level.
- *Social:* We anticipate MBDO to increase transparency for factory stakeholders as it can make (i) information about factory operation available in a concern-oriented, model-based fashion; and (ii) design deviations apparent.

III. Model-Based DevOps: Foundations and Challenges

Models can be leveraged throughout all phases of the software engineering process [10]. When considering the two dimensions of DevOps, *design models* target the Dev dimension and describe, e.g., requirements, architecture components, and functional concerns. Code generation from such models can increase development efficiency, but also agility due to facilitated refactoring. *Operation models* support the Ops dimension by specifying, e.g., component configurations and deployments. Following Models@run.time [11], operation models of application runtime can support the reconfiguration of running systems from model adaptations [12]. Such *runtime models* are thus primitive digital twins of running systems [13].

While both Dev and Ops dimensions can benefit from MDE, in traditional DevOps the corresponding models are typically separated. Consequently, there is no direct impact of the monitored system state on its design models. Software engineers are thus required to implement, test, and redeploy design adaptations after recognizing errors in the system state. In this case, there only exists a manual feedback loop from Ops to Dev dimension. Self-adaptive systems [14] aim to automate this feedback loop by means of a reasoning engine and reflection layer that can also be modeled [12], [15]. However, adaptability is then usually limited to very specific aspects like web server provisioning and has to be designed upfront. In addition, there is often no direct connection from the Dev to the Ops dimension, and thus no relationship between design and operation models, even though model transformations could aid in deriving operation from design models [10]. Furthermore, when retrofitting existing systems like CPSs with MBDO capabilities (Sect. II), the generation of runtime models from engineering models and state data represented by, e.g., log files or databases, becomes crucial to allow model-based monitoring.

With MBDO as a novel research avenue, we aim to strengthen the link between Dev and Ops models by a *generic* framework that (i) considers the heterogeneity of design and operation models, yet supports the expression of their connection in a unifying manner; (ii) provides means to (semi-) automatically generate digital twins of modeled systems including connections between design and operation models on the implementation level; and (iii) enables flexible navigation between the models. In particular, MBDO focuses on the bidirectional relationship between design and operation models, and the systematic derivation of Dev-to-Ops-to-Dev models. Hence, it improves on the current state-of-research of integrating MDE and DevOps that leverages MDE technologies for the development of DevOps platforms and thus forward engineering alone [16].

The adoption of MBDO in the context of our demonstrator (Sect. II) enables the generation of (i) a factory module's digital twin from its design model; and (ii) a factory digital twin as

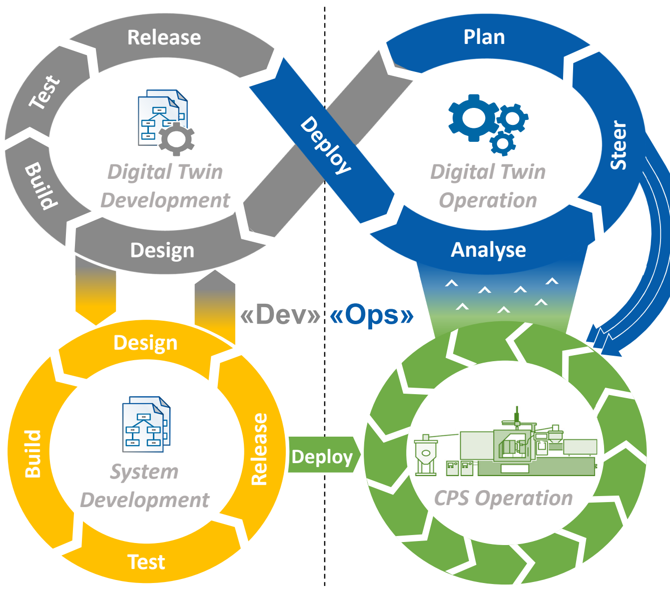

Figure 2: Envisioned impact of MBDO on system development and CPS operation.

a composition of module twins. However, MBDO also allows reflecting runtime model changes back to the design models of digital twins, thereby enabling sustainability improvements as described in Sect. II. Fig. 2 summarizes our vision of the MBDO research avenue: Digital twins originate from and impact system development (Dev), capture the system state (Ops), and allow navigation between both dimensions.

We see three major challenges in approaching MBDO:

a) Challenge 1: Controlled Updates of Runtime Models: This challenge concerns the self-adaptation of CPSs. Open research questions in this context comprise (i) how can capturing runtime data be expressed and configured in a generic, model-based fashion; (ii) how can a traceable connection between design and operation models be established; and (iii) how can updates of runtime models for self-adaptation be controlled?

b) Challenge 2: Digital Twins for DevOps Integration: For MBDO, it is natural to leverage digital twins as integration mechanisms that relate design and operation models because digital twins can store, process, and manage runtime data efficiently [17]. However, it is currently unclear (i) how digital twins maintain links between data models that can impact each other; and (ii) how digital twins can be leveraged as information and control centers to manage connections of design and operation models.

c) Challenge 3: Automated Dev-to-Ops-to-Dev: MBDO's main goal is the derivation of Dev-to-Ops-to-Dev models that enable flexible navigation between design and operation models, and thus reify runtime impact on system design. This challenge requires reasoning about (i) identifying deviations between system design and system state in a generic fashion; (ii) equipping digital twins (Challenge 2) with support for Dev-to-Ops-to-Dev model derivation; and (iii) leveraging digital twins to assist CPS development.

IV. Research Roadmap

This section presents our roadmap to examine MBDO as a novel research avenue. Each subsection focuses on a step in the roadmap and relates to the challenges from Sect. III.

A. Composability of Digital Twins

MBDO focuses on digital twins that reflect connections between design and operation models (Sect. III). Since CPSs commonly involve many such models, there likely is a variety of MBDO-induced digital twins, which poses the question of their composability and how MBDO's opportunities for continuous improvement impact composed digital twins.

In general, we consider a hierarchical approach for the composition of *sub-component digital twins* to more complex digital twins. When using a gateway to provide access to digital twins, the composition of digital twins incurs a composition of their gateways—whether for one/multiple or same/different CPSs. Gateways then ensure digital twin interoperability. Next to the question how MBDO can be applied to digital twins that are logically or physically composed, its impact on gateway-based digital twin compositions also requires investigation.

We attempt to answer these questions and show the feasibility of the MBDO approach on our demonstrator (Sect. II), which consists of multiple modules with digital twins and thus comprises a composed digital twin.

B. Definition of a Reference Model of Digital Twins and Data Models

This step aims to establish a generic foundation to connect digital twins with data models (Challenge 2 in Sect. III). We reify this foundation as a reference model that conceptualizes our common understanding of the core elements of digital twins and data models, and, specifically, the relationship between them. The reference model will enable us to leverage existing insights and methodologies for digital twins and data models in a combined fashion, making it possible to employ digital twins as mediators between different models of CPS runtime data.

C. Enabling the Modeling of Ops Activities

This step addresses Challenge 1 (Sect. III), and focuses on the design and implementation of model-based means for the representation of runtime data in the Ops phase. To this end, we first devise a conceptual framework for runtime data definition and subsequently a Domain-Specific Language (DSL) for this purpose. A core feature of this DSL will be the model-based expression of associations between captured runtime data and digital twins. In addition, we plan to study transformations of monitored state data to DSL models.

Starting from the aforementioned conceptual framework, we investigate the development of a system to query runtime data. This query system shall enable the extraction of information about past, present, and future states of CPSs governed by digital twins and their associated operation models. Conceptually, the query system establishes the linkage between models created during the design phase of a CPS, e.g., state machine models, and their runtime representation, e.g., the history of reached states. It is thus crucial to MBDO.

The differentiation between design and runtime models is primarily driven by the level of abstraction inherent to the model-based design of digital twins and the possibility of encountering unforeseen states during CPS runtime. Consequently, we will also have to consider the safe handling of self-adaptation of design models during runtime.

D. Enabling the Modeling of Dev Activities

This step is closely related with the previous one and focuses on investigating the modeling aspects of activities in the Dev phase. For this purpose, we can draw on a significant body of knowledge resulting from intense research activities concerning the engineering of digital twins with MDE principles and techniques. We plan to capture this knowledge in the form of a DSL for Dev activities in model-based digital twin engineering and integrate this DSL with the one for runtime data. As a result, we can effectively address and explore Dev activities in MBDO w.r.t. the Ops phase (Sect. IV-C) and Challenge 2 (Sect. III). An interesting question will be the effective handling of models that were created from different perspectives and development teams.

E. Establishing the Dev-to-Ops-to-Dev Modeling Continuum

This step tackles Challenge 3, Part (i) in that it establishes the foundations for a continuum between Dev and Ops. This continuum relies on the DSLs for Ops and Dev activities, and their integration, which we expect to support the automated synthesis of Dev-to-Ops-to-Dev models (Sect. IV-F).

However, the step also concerns Challenge 2 as our demonstrator's digital twins (Sect. II) will also provide control centers that incorporate Dev-to-Ops-to-Dev information and make it available to end users.

F. Automatically Synthesizing Dev-to-Ops-to-Dev Models

This step addresses Challenges 3, Part (ii) to automatically synthesize Dev-to-Ops-to-Dev models. For this purpose, we draw on existing transformation frameworks to specify and execute model transformations from engineering and design models, including information about machine structure, geometry, kinematics, and behavior, to operation models. We apply these models to our demonstrator (Sect. II) to subsequently improve MBDO's conceptual frameworks and solution artifacts. Therefore, it is important to automatically identify deviations between the demonstrator's design and current state, allowing for design model updates that reflect the system state accurately. Ultimately, this would enable prompt adjustments and improvements in response to changing production conditions.

G. Exploration of Further Research Opportunities for MBDO

Finally, we explore further opportunities to study MBDO and derive future research questions (Challenge 3, Part (iii)).

For example, we anticipate the provisioning of a framework for linters that are dedicated to quality-of-service properties of CPSs under MBDO control. These linters could assist developers at Dev time with estimations about Ops time properties. Hence, they would particularly help developers to better understand a system's behavior prior to its deployment and also make it possible to evaluate different deployment strategies already at Dev time.

V. Related Work

We present work related to MBDO w.r.t. model-driven engineering for Industry 4.0 (Sect. V-A), digital twins (Sect. V-B), and DevOps (Sect. V-C).

A. Model-Driven Engineering for Industry 4.0

Bellavista et al. [18] present a conceptual architecture for a digital twin of Industry 4.0 factories. The twin is represented as a containerized service with several interfaces, each of which providing specific functionalities to physical machines or other digital services. The digital twin service relies on a model that covers its state, design, and behavior. The presented architecture employs several software design patterns to describe (i) the relations of the twin service's components; (ii) the relations between the service and its external environment; and (iii) the twin's adaptability to various contexts. While this approach proposes a concept for digital twin implementation, other than MBDO it neither considers the impact of runtime data on the twin's design model nor the derivation of operation models from runtime data.

B. Model-Driven Engineering for Digital Twins

There exist some MDE approaches which support the engineering of digital twins with generative methods. However, none of these aim to cover the whole lifecycle of original systems, and the relationships between Dev and Ops time using models.

[19] uses MDE for creating digital twin architectures based on architecture description languages. [20] proposes a framework for MDE and maintenance of digital twin infrastructures based on AutomationML[2] models. [21] introduces a modeling language and code generation framework for the monitoring of mobile CPS. [13] proposes an approach to generate interfaces between a CPS and its digital twin. Focusing on digital twin visualization functionalities, [22] describes how to derive digital twin cockpits from models and [17] introduces an approach to derive process-aware digital twin cockpits from event logs.

C. Model-Driven Engineering for DevOps

Adapting models to the context, particularly during runtime, is one of the challenges of digital twins [23]. However, they are either dedicated to pure software architecture models, rather than CPSs, or do not recognize the Dev-to-Ops-to-Dev continuum. Szvetits and Zdun [24] conducted a comparison of existing work for the construction of runtime models. mRUBIS [25] provides runtime self-healing and self-optimization capabilities for software architectures but is not dedicated to digital twins. Muñoz et al. [26] proposed a method to align

[2] https://www.automationml.org

the traces produced by original systems with those produced by their digital twins, enabling the runtime to evaluate twins' consistency with the real world. Hughes et al. [27] devised an approach that leverages DevOps to build digital twins and incorporate feedback for model improvement. However, runtime models can only be improved through the construction of a new model that reflects Ops feedback. There is no construction of multiple models in Dev and Ops phases with a continuous evaluation of their deviation at runtime. Raţiu et al. [28] introduce the notion of *reactive links* to connect properties of multi-domain engineering models and propagate property changes between models as they occur based on a set of predefined actions. We consider reactive links a potential foundation for expressing links between design and operation models. However, we expect MBDO to require the specification of more complex and coarse-grained actions for change propagation and the corresponding triggers.

VI. Conclusions and Future Work

This paper presented Model-Based DevOps — a novel research avenue that connects the design models of cyber-physical systems with their runtime monitoring in the form of digital twins. We expect MBDO to benefit the sustainability of factories as it enables to (i) increase production process efficiency from simulated reconfigurations; (ii) assess and optimize resource consumptions and factory emissions w.r.t. product quality and process efficiency; and (iii) increase transparency for stakeholders in a concern-oriented fashion. Starting from a case study demonstrator for sustainable Industry 4.0 (Sect. II), we identified foundations and major challenges for MBDO's investigation (Sect. III), and derived a corresponding research roadmap (Sect. IV).

In future work, we approach this roadmap for a principled and rigorous investigation of MBDO, its theoretical framework and practical implications for CPS design and operation.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Model-Based DevOps – 505496753. Website: https://mbdo.github.io.

References

[1] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect's Perspective*. Pearson, 2015.

[2] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[3] R. Eramo, F. Bordeleau, B. Combemale, M. v. d. Brand, M. Wimmer, and A. Wortmann, "Conceptualizing digital twins," *IEEE Software*, vol. 39, no. 2, pp. 39–46, 2022.

[4] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, 2014.

[5] M. Heithoff, A. Hellwig, J. Michael, and B. Rumpe, "Digital twins for sustainable software systems," in *2023 IEEE/ACM 7th Int. Workshop on Green And Sustainable Software (GREENS)*, 2023, pp. 19–23.

[6] 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*. Springer, 2023, pp. 3–31.

[7] S. Fur, M. Heithoff, J. Michael, L. Netz, J. Pfeiffer, B. Rumpe, and A. Wortmann, *Sustainable Digital Twin Engineering for the Internet of Production*. Springer, April 2023, pp. 101–121.

[8] I. 23247-1, "Automation systems and integration—digital twin framework for manufacturing—part 1: overview and general principles," 2020.

[9] B. Giddings, B. Hopwood, and G. O'brien, "Environment, economy and society: fitting them together into sustainable development," *Sustainable development*, vol. 10, no. 4, pp. 187–196, 2002.

[10] B. Combemale, R. France, J.-M. Jézéquel, B. Rumpe, J. Steel, and D. Vojtisek, *Engineering Modeling Languages: Turning Domain Knowledge into Tools*. Chapman & Hall, 2016.

[11] G. Blair, N. Bencomo, and R. B. France, "Models@run.time," *Computer*, vol. 42, no. 10, pp. 22–27, 2009.

[12] B. Morin, O. Barais, J.-M. Jezequel, F. Fleurey, and A. Solberg, "Models@run.time to support dynamic adaptation," *Computer*, vol. 42, no. 10, pp. 44–51, 2009.

[13] J. C. Kirchhof, J. Michael, B. Rumpe, S. Varga, and A. Wortmann, "Model-driven Digital Twin Construction: Synthesizing the Integration of Cyber-Physical Systems with Their Information Systems," in *23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems*. ACM, 2020.

[14] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, "A survey on engineering approaches for self-adaptive systems," *Pervasive and Mobile Computing*, vol. 17, pp. 184–206, 2015.

[15] M. Amin, "Toward self-healing infrastructure systems," *Computer*, vol. 33, no. 8, pp. 44–53, 2000.

[16] M. Ugarte Querejeta, L. Etxeberria, and G. Sagardui, "Towards a devops approach in cyber physical production systems using digital twins," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*. Springer, 2020, pp. 205–216.

[17] D. Bano, J. Michael, B. Rumpe, S. Varga, and M. Weske, "Process-Aware Digital Twin Cockpit Synthesis from Event Logs," *Journal of Computer Languages*, vol. 70, June 2022.

[18] P. Bellavista, N. Bicocchi, M. Fogli, C. Giannelli, M. Mamei, and M. Picone, "Requirements and design patterns for adaptive, autonomous, and context-aware digital twins in industry 4.0 digital factories," *Computers in Industry*, vol. 149, p. 103918, 2023.

[19] 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 *Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, vol. 12127. Springer, 2020.

[20] D. Lehner, S. Sint, M. Vierhauser, W. Narzt, and M. Wimmer, "AML4DT: A model-driven framework for developing and maintaining digital twins with automationml," in *2021 26th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–8.

[21] A. Fend and D. Bork, "CPSAML: A language and code generation framework for digital twin based monitoring of mobile cyber-physical systems," in *Int. Conf. on Model Driven Engineering Languages and Systems: Companion Proceedings*, ser. MODELS '22. ACM, 2022.

[22] M. Dalibor, J. Michael, B. Rumpe, S. Varga, and A. Wortmann, "Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits," in *Conceptual Modeling*. Springer, 2020, pp. 377–387.

[23] T. Brockhoff, M. Heithoff, I. Koren, J. Michael, J. Pfeiffer, B. Rumpe, M. S. Uysal, W. M. P. van der Aalst, and A. Wortmann, "Process Prediction with Digital Twins," in *Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. ACM/IEEE, 2021.

[24] M. Szvetits and U. Zdun, "Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime," *Softw. Syst. Model.*, vol. 15, no. 1, pp. 31–69, 2016.

[25] T. Vogel, "mRUBiS: an exemplar for model-based architectural self-healing and self-optimization," in *13th Int. Conf. on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. ACM, 2018.

[26] P. Muñoz, M. Wimmer, J. Troya, and A. Vallecillo, "Using trace alignments for measuring the similarity between a physical and its digital twin," in *25th Int. Conf. on Model Driven Engineering Languages and Systems: Companion*. ACM, 2022, pp. 503–510.

[27] J. Hugues, A. Hristosov, J. J. Hudak, and J. Yankel, "TwinOps - DevOps Meets Model-Based Engineering and Digital Twins for the Engineering of CPS," in *23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems: Comp.* ACM, 2020.

[28] C. C. Raţiu, W. K. G. Assunção, R. Haas, and A. Egyed, "Reactive Links across Multi-Domain Engineering Models," in *25th Int. Conf. on Model Driven Engineering Languages and Systems (MODELS)*. ACM, 2022, pp. 76–86.