Engineering Digital Process Twins: Towards a Methodology Based on MDE and DevOps

Gwendal Beaumont*
Department of Computer Sciences
IMT Atlantique
Brest, France
gwendal.beaumont@imt-atlantique.fr

Charlotte Verbruggen
Business Informatics Group
TU Wien
Vienna, Austria
charlotte.verbruggen@tuwien.ac.at

Antoine Beugnard
Department of Computer Sciences
IMT Atlantique
Brest, France
antoine.beugnard@imt-atlantique.fr

Dominik Bork
Business Informatics Group
TU Wien
Vienna, Austria
dominik.bork@tuwien.ac.at

Salvador Martínez
Department of Computer Sciences
IMT Atlantique
Brest, France
salvador.martinez@imt-atlantique.fr

Christelle Urtado
SyCoIA
IMT Mines Ales
Ales, France
Christelle.Urtado@mines-ales.fr

Sylvain Vauttier
SyCoIA
IMT Mines Ales
Ales, France
sylvain.vauttier@mines-ales.fr

Abstract

A Digital Process Twin (DPT) is a digital twin (DT) of a business process designed to monitor, adapt, and optimize organizational workflows to maintain the efficiency, adaptability, and competitiveness of businesses. The benefits of DPTs are many: outcome prediction, data-driven insights for optimization, and streamlined tracking of inefficiencies. However, developing DTs of business processes introduces new challenges, such as a lack of tool support for DPTs, interoperability challenges between DPTs, and integration of process assets such as user input and external databases. To address these challenges, this paper presents an initial framework for DPTs deployment, illustrated through a library management system use case. Our approach leverages MERODE, a Model-Driven Engineering (MDE) approach for developing enterprise information systems, and GitOps, an application of DevOps principles to Git-based version control, to automate the static description of business assets.

CCS Concepts

• **Information systems → Enterprise information systems**; • **Software and its engineering → Software development techniques**.

Keywords

Business Processes, DevOps, Digital Twins, Digital Process Twins

ACM Reference Format:
Gwendal Beaumont, Charlotte Verbruggen, Antoine Beugnard, Dominik Bork, Salvador Martínez, Christelle Urtado, and Sylvain Vauttier. 2026. Engineering Digital Process Twins: Towards a Methodology Based on MDE and DevOps. In *ACM/IEEE 29th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion 2026), October 04–09, 2026, Málaga, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3837062.3839100

1 Introduction

Modern organizations face intense pressure to continuously improve the cost-efficiency of their activities to remain competitive [17]. Over the past decades, advances in the Internet-of-Things (IoT) and real-time sensor data acquisition have transformed how organizations monitor and optimize their business processes (BPs) [17, 26]. These IoT-enhanced BPs integrate cyber-physical systems with sensing and actuation capabilities that improve operational efficiency. The benefits are many, including reduced human error through automation and better visibility of operations, which help make more informed decisions [17]. As key components of the digital twin (DT) paradigm, IoT devices have enabled the construction of DTs that mirror BPs and actuate on them, also known as Digital Process Twins (DPTs) [12, 17, 23]. DPTs have the potential to support business competitiveness by enabling data-driven decision-making, facilitating early prediction, mitigating costly errors, and optimizing workflows and operational efficiency through simulation and automation [26].

However, Fornari et al. [17] note that merging the DT paradigm with BPs is far from trivial and identify challenges in the development of DPTs. They highlight a *lack of software tools engineered*

* Also with SyCoIAIMT Mines Ales,.

ACM ISBN 979-8-4007-2903-4/26/10
https://doi.org/10.1145/3837062.3839100

for DPTs and the difficulties in scaling DPTs to DTs of complete organizations (i.e., composing multiple DPTs to form the DT of an organization). Furthermore, while traditional DTs usually operate within structured environments, DPTs must integrate new, disparate types of assets (e.g., employee input, printed documents, external databases) [17]. Research contributions on the representation of DPTs also exist [30]; however, they often lack a systematic approach for the implementation and deployment of DPTs [13, 17]. In light of these existing contributions, **the objective of this paper is to propose a methodology for the systematic implementation and deployment of digital process twins (DPTs)**.

To achieve this, the proposed methodology is based on two pillars: DevOps and the MERODE methodology. The former has emerged as a successful approach to managing the entire development life cycle of a software system, from its design phase to its end of life [1, 6]. The latter is a model-driven engineering (MDE) approach supporting the automated generation of web applications. These applications cover data management and behavior of an information system based on a MERODE domain model. This integration aligns with a growing number of studies that back the use of DevOps principles in DT applications [4, 9, 16, 20, 25]. Integrating these two approaches is a first step towards enhancing the level of automation of the design, development, and deployment of DPTs. To illustrate our proposal, we use a library management system use case.

In the remainder of this paper, Section 2 introduces the concepts that ease the understanding of the proposed approach and contrasts this work with existing literature; Section 3 describes the library use case; Section 4 details our approach and showcases a preliminary implementation; and Section 5 concludes the paper and outlines future work directions.

2 Background & Related Works

2.1 MERODE Methodology

The MERODE methodology is an MDE approach that uses object-oriented domain modeling to develop enterprise information systems [10, 28]. It comprises three layers: the *Domain* layer, the *Information System Services* layer, and the *Business Process* layer [28].

The **Domain layer** defines *business objects*, including their attributes and relationships [27]. MERODE enables code generation from the *MERODE Domain Model.* A MERODE domain model (as illustrated in Fig. 1) is composed of three views: an *Existence Dependency Graph (EDG)* which defines relationships between entities, an *Object Event Table (OET)* which presents the relationships between actions and entities, and a set of *Finite State Machines (FSMs)* for each entity [27]. The **Information System Services (ISS) layer** is the bridge between business objects and BPs [27]. Input services update business objects while output services provide access to the data. Finally, the **Business Process layer** facilitates interactions between processes and the domain layer via the ISS [27].

While MERODE is traditionally used for enterprise information systems, recent related work has begun exploring its integration with the DT paradigm [10, 11, 17]. Building on these modeling capabilities, our approach applies MERODE to the business layer of the DPT. Specifically, Compagnucci et al. presented DOLLY, a MERODE-based framework for DT specification in the IoT domain [10, 11]. In this framework, the MERODE model is predefined to describe IoT devices and cannot be changed for another MERODE model (e.g., a library information system). Our approach aims to go further by enabling the deployment of DPTs regardless of the chosen MERODE model.

2.2 Model-driven engineering applied to DTs

Existing approaches in the literature leverage model-driven engineering (MDE) to develop DTs [5, 7, 8, 15, 19]. Recently, Lehner et al. [22] and Michael et al. [25] published a systematic literature review and a perspective paper on the merging of MDE and DTs. In their previous work, Lehner et al. [21] propose a reference architecture to take advantage of the principles of MDE to connect models crafted at design time with running systems [21]. It proposes using model repositories as artifacts from the design phase, connecting them to running systems, and using them as a source of information. Other propositions from Bibow et al. [7], Binder et al. [8], Dalibor et al. [15], and Hugues et al. [19] propose MDE-based architectures to develop DTs of CPSs in varying contexts (e.g., Industry 4.0 and IoT). While they all successfully describe physical assets (e.g., sensors, machines), none is tailored for integrating business processes. This paper differentiates itself by applying MDE principles to a currently underexplored class of DTs: Digital Process Twins (DPTs).

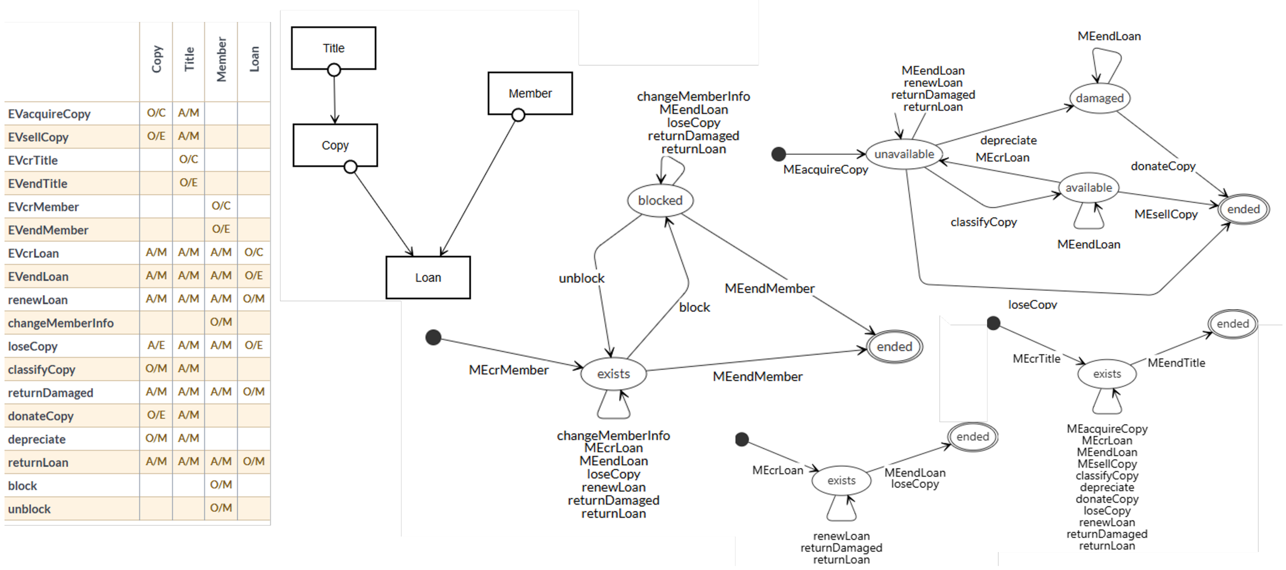
2.3 DevOps, GitOps and DTs

The DevOps approach enables the continuous evolution of software by improving collaboration between development (Dev) and operations (Ops) teams. It has emerged as a successful approach for managing complexity in regular software applications [2, 24]. Recently, interest in applying DevOps to DTs has grown [3, 9, 16, 19, 25]. GitOps applies DevOps principles to Git, a version control software [4]. Thanks to an ecosystem of tools (e.g., GitLab, ArgoCD, Kubernetes) and the benefits of DevOps, GitOps enables a version-controlled, specific deployment configuration. It also holds other promises such as faster deployments, easy rollback to previous states, and a tool-independent architecture [18].

Existing frameworks apply GitOps to systematically engineer and evolve DTs [1]. For instance, Koren et al. [20] introduce the DevOps4CPPS framework, which highlights the use of microservices and GitOps in a Cyber-Physical System environment. Similarly, recent platforms proposed by Talasila et al. [29] and Dalibor et al. [14] leverage GitOps to execute and manage DTs. Our prior research [4] applies GitOps to DT development by proposing a DT lifecycle and mapping GitOps operations onto its transitions. By generating configuration artifacts that can directly be used by GitOps tools, we established a baseline for GitOps-driven DT deployment. The approach presented in this paper builds upon that foundation to systematically deploy DPTs.

3 Library Use Case

To illustrate our approach, we detail a use case (UC) involving a library management system. The library management system comprises four main entities: *Title*, *Copy*, *Member*, and *Loan*. Titles and members can be created without restrictions, while copies require a title and loans require a copy and a member. The event *EVacquireCopy* creates a new record of a *copy* in the system. Copies



	Copy	Title	Member	Loan
EVacquireCopy	O/C	A/M		
EVsellCopy	O/E	A/M		
EVcrTitle		O/C		
EVendTitle		O/E		
EVcrMember			O/C	
EVendMember			O/E	
EVcrLoan	A/M	A/M	A/M	O/C
EVendLoan	A/M	A/M	A/M	O/E
renewLoan	A/M	A/M	A/M	O/M
changeMemberInfo			O/M	
loseCopy	A/E	A/M	A/M	O/E
classifyCopy	O/M	A/M		
returnDamaged	A/M	A/M	A/M	O/M
donateCopy	O/E	A/M		
depreciate	O/M	A/M		
returnLoan	A/M	A/M	A/M	O/M
block			O/M	
unblock			O/M	

Figure 1: The MERODE model for the Library case. From left to right: the Object Event Table (OET), the Existence Dependency Graph (EDG), and the Finite-State Machines (FSM) for `Member`, `Loan`, `Copy`, and `Title`.

become available after a librarian has classified them (*classifyCopy*). When a copy is on loan, it becomes unavailable until it is classified again. A copy that is too damaged can be *depreciated* by the librarian and will be *donated*. Books that are still in good condition but not very popular can be *sold*. Members who damage or lose copies can be *blocked* by the librarian. They are not allowed to take out loans until they are unblocked.

Following the description, we model the UC and generate associated artifacts using the MERODE tools[1]. Fig. 1 details the library EDG in the MERODE notation. The OET on the left part of Fig. 1 presents the relations and dependencies between events and the different entities of the library. Each event can be linked to different entities using an X/Y notation where: X refers to the participation of the entity in the event (O = owned, A = acquired), and Y refers to the kind of event (C = create, M = modify, E = end). Fig. 1 also details the Member, Loan, Copy, and Title FSMs of the UC.

4 Proposed solution

4.1 Description of the approach

Our approach consists of 1) **instantiating the GitOps model proposed in prior work** [4] and 2) **generating Node-RED workflows through a text-to-text transformation based on the MERODE Domain model**. The goal here is to systematically deploy the right artifacts and services from a given platform and BP configuration to power our DPT.

First, we instantiate a GitOps model (see Fig. 3) of the necessary elements to deploy a working DPT. These include a `GitOpsPlatform` which represents the platform we want to model, an `Artifact-Registry` that stores artifacts, a `GitRepository` that stores the target state of the platform, a `ContainerImageRecipe` to build ready-to-use versions of the associated `ContainerImage` (i.e., versions of the services), and an `OrchestrationRecipe` with its associated `DeploymentServices` and `DeploymentNetwork` to describe how are containers are deployed, connected, and scaled across a cluster. This step allows to systematically deploy the necessary infrastructure to support the realization of our DPT.

Then, to enable process execution in the deployed instance, we use Node-RED (`dashboard:DeploymentService` in Fig. 3), a low-code programming platform based on *node-based* workflows (represented in a JSON file). Each node corresponds to a specific functionality (e.g., an *http request* node supports making HTTP requests). Our approach proposes to systematically **generate Node-RED workflows via a text-to-text transformation based on the MERODE Domain model**. The MERODE model (represented as an XML-based file) provides information about the components that need to be included in the Node-RED environment. For *each object type in the EDG*, the *dashboard should contain a tab* with the following components: a table visualizing the instances of the object type, including attributes and a button to remove (EVend) the instance, and a field to create new instances of the object type, where attribute values and associations to instances of other object types can be specified.

Fig. 2 illustrates an overview of the proposed solution. After instantiating the GitOps model, the business modeler manually crafts the MERODE domain model, which serves as an input to both the Node-RED generator and the MERODE interpreter. The Node-RED generator generates Node-RED workflows while the MERODE interpreter instantiates a MERODE live model. The business modeler can manually add more advanced workflows to Node-RED. Then, by clicking on a “Deploy” button, the user triggers the automatic generation of the interface that the business user can interact with.

[1]KU Leuven, MERLIN, accessed June 21, 2026, https://merlin-academic.com/

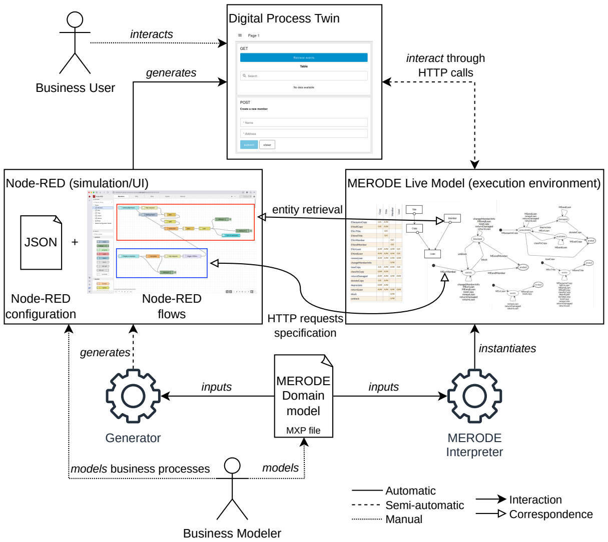


Figure 2: Overview of the proposed approach

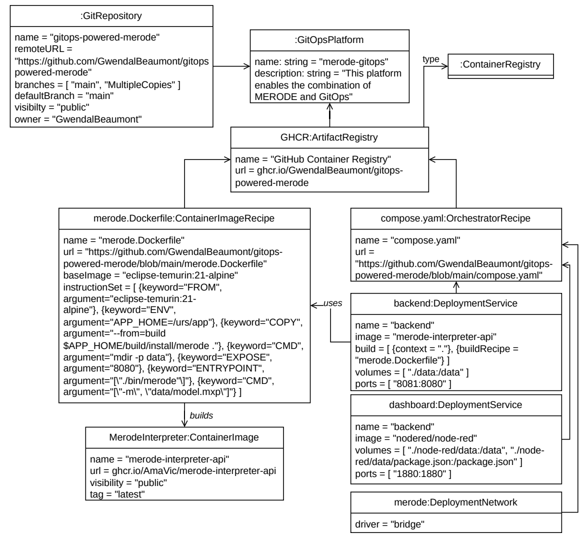


Figure 3: Prototype instantiation model of our platform

4.2 Proof-of-concept implementation

We present a proof-of-concept implementation of the approach. First, we instantiate the GitOps model. Fig. 3 depicts our prototype instantiation including a `GitOpsPlatform`, the central point of the platform, linked to a single `GitRepository` hosted on GitHub[2]. The repository contains the artifacts required for deploying the DPT. Artifacts, such as the `MerodeInterpreter` (of class `ContainerImage`) can be uploaded directly to the GitHub Container Registry (GHCR) (`ArtifactRegistry` class). Our platform contains two other artifacts: a `merode.Dockerfile` (`ContainerImageRecipe` class) describing how to build the `MerodeInterpreter`, and a `compose.yaml` (`OrchestratorRecipe` class) describing the different `Deployment-Services` and `DeploymentNetwork` required by the DPT.

Using the *Member* object type as an example, two components are specified in Node-RED using flows. The first component (red box in Fig. 2) can be modeled in a flow with three steps. The first step is a button (*retrieveMembers*) that triggers an HTTP request to the MERODE interpreter. The second step parses the HTTP request response to extract the information needed to display the *members*. The third step displays *members* in a table. The second component (blue box in Fig. 2) is a field for creating new instances. This component contains a form in which users can enter attribute values and relations to other instances (*Create a member*). Then, a template is used transform the input into a JSON structure that the MERODE interpreter can read and send via an HTTP request.

The combination of the GitOps instantiated model and the Node-RED workflows generator should allow us to provide business users with a graphical interface that enables direct interaction with the DPT. In our library use case example, we use Docker — a common GitOps containerization tool — to execute the instantiated model of our GitOps platform. Then, the Node-RED generator reads the MERODE domain model to generate Node-RED workflows. These workflows should represent actual events from the MERODE OET and FSMs. Once generated, the user connects to Node-RED's administration interface and can a) deploy the interface 'as-is' and start interacting with the DPT or b) manually add custom, more advanced processes before interacting with the DT.

[2]https://github.com/GwendalBeaumont/gitops-powered-merode

5 Conclusion & Future Works

This paper presented an approach towards supporting the development and deployment of Digital Process Twins using a combination of the MERODE methodology and the GitOps framework. Using this approach, we are able to instantiate the DPT of a library management system and propose the premises of a monitoring dashboard. We detailed how we map MERODE artifacts such as the model XML file to GitOps elements, and asserted which of these mappings are manual, semi-automatic or automatic. With the library use case, we validated that the semi-automatic mappings can be fully automated in the future by defining model transformations. This will allow the business modeler to develop and deploy DPTs using Node-RED flows and a MERODE domain model, without having to be concerned about implementing the interaction between the systems. Our solution is a first step towards addressing the challenges identified [17, 25] by integrating MDE and DevOps methodologies and providing (semi-)automated mappings and transformations between the models. The artifacts of our approach are publicly available to the community[2].

As future works, we intend to extend this approach to provide additional services to final users embedded directly into the dashboard. These includes more advanced workflows such as cascading (streamlining) and composing (parallelizing) events. Additionally, we intend to integrate new types of artifacts not provided by the MERLIN tool but still part of the MERODE methodology such as BP models, usually represented using the BPMN notation [27]. Including BPMN models and providing a (semi-)automated transformation from BPMN models to Node-RED flows will support the Business Modeler to include humans interactions with the system into the DPT development.

Acknowledgments

Gwendal Beaumont's Ph.D. is funded by the Futur & ruptures program by Fondation Mines-Télécom and the TSN CARNOT institute. Authors also thank the EULiST Alliance for funding the mobility grant that made this collaborative research possible.

References

[1] Sara Aissat, Jonathan Beaulieu, Érik Poirier, Ali Motamedi, Julien Gascon-Samson, and Francis Bordeleau. 2025. A devops framework for the systematic engineering and evolution of digital twins for built assets. *Software and Systems Modeling* 25, 3 (Oct. 2025), 815–845. doi:10.1007/s10270-025-01330-0

[2] Anand Srivatsav Amaradri and Swetha Bindu Nutalapati. 2016. *Continuous Integration, Deployment and Testing in DevOps Environment.* Master's thesis. Blekinge Institute of Technology, Karlskrona, Sweden. http://bth.diva-portal.org/smash/get/diva2:1044691/FULLTEXT02.pdf

[3] Gwendal Beaumont, Antoine Beugnard, Jannik Laval, Salvador Martínez, Souad R. Chaniour, Christelle Urtado, Sylvain Vauttier, and Bérenger Akodo. 2026. DevOps for and by Digital Twins — Leveraging Virtual Replicas in Continuous Software Engineering. In *Digital Twins*, Bedir Tekinerdogan and Cor Verdouw (Eds.). Elsevier B.V., 227–242. doi:10.1016/B978-0-443-45573-5.00018-5

[4] Gwendal Beaumont, Antoine Beugnard, Salvador Martínez, Christelle Urtado, and Sylvain Vauttier. 2025. Towards Automating the Life Cycle Management of Digital Twins. In *ER 2025*, Dominik Bork, Roman Lukyanenko, Shazia Sadiq, Ladjel Bellatreche, and Óscar Pastor (Eds.), Vol. 16189. Springer Nature Switzerland, Poitiers, France, 412–430. doi:10.1007/978-3-032-08623-5_22

[5] Gaëlic Béchu, Antoine Beugnard, Caroline G. L. Cao, Quentin Perez, Christelle Urtado, and Sylvain Vauttier. 2022. A Software Engineering Point of View on Digital Twin Architecture. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, Stuttgart, Germany, 1–4. doi:10.1109/etfa52439.2022.9921617

[6] Luca Berardinelli, Vittoriano Muttillo, Romina Eramo, Hugo Bruneliere, Abbas Rahimi, Antonio Cicchetti, Joan Giner-Miguelez, Abel Gómez, Pasqualina Potena, and Mehrdad Saadatmand. 2025. Model Driven Engineering, Artificial Intelligence, and DevOps for Software and Systems Engineering: A Systematic Mapping Study of Synergies and Challenges. *ACM Transactions on Software Engineering and Methodology* 35, 8 (Aug. 2025), 3759454. doi:10.1145/3759454

[7] Pascal Bibow, Manuela Dalibor, Christian Hopmann, Ben Mainz, Bernhard Rumpe, David Schmalzing, Mauritius Schmitz, and Andreas Wortmann. 2020. Model-Driven Development of a Digital Twin for Injection Molding. In *Advanced Information Systems Engineering*, Schahram Dustdar, Eric Yu, Camille Salinesi, Dominique Rieu, and Vik Pant (Eds.). Springer International Publishing, Cham, 85–100. doi:10.1007/978-3-030-49435-3_6

[8] Christoph Binder, Ambra Cala, Jan Vollmar, Christian Neureiter, and Arndt Luder. 2021. Automated Model Transformation in Modeling Digital Twins of Industrial Internet-of-Things Applications Utilizing AutomationML. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, Vasteras, Sweden, 1–6. doi:10.1109/ETFA45728.2021.9613172

[9] Benoit Combemale and Manuel Wimmer. 2020. Towards a Model-Based DevOps for Cyber-Physical Systems. In *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, Jean-Michel Bruel, Manuel Mazzara, and Bertrand Meyer (Eds.). Springer International Publishing, Cham, 84–94. doi:10.1007/978-3-030-39306-9_6

[10] Ivan Compagnucci, Barbara Re, Estefanía Serral Asensio, and Monique Snoeck. 2025. A Digital Process Twin Conceptual Architecture for What-If Process Analysis. In *Enterprise Design, Operations, and Computing. EDOC 2024 Workshops*, Monika Kaczmarek-Heß, Kristina Rosenthal, Marek Suchánek, Miguel Mira Da Silva, Henderik A. Proper, and Marianne Schnellmann (Eds.). Vol. 537. Springer Nature Switzerland, Cham, 373–388. doi:10.1007/978-3-031-79059-1_23

[11] Ivan Compagnucci, Monique Snoeck, and Estefanía Serral Asensio. 2023. Supporting Digital Twins Systems Integrating the MERODE Approach. In *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, Västerås, Sweden, 449–458. doi:10.1109/MODELS-C59198.2023.00079

[12] Flavio Corradini, Sara Pettinari, Barbara Re, Lorenzo Rossi, and Francesco Tiezzi. 2022. An Approach to Support Digital Process Twin. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, Falerna, Italy, 1–4. doi:10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9927841

[13] Flavio Corradini, Sara Pettinari, Barbara Re, Lorenzo Rossi, and Francesco Tiezzi. 2023. Executable Digital Process Twins: Towards the Enhancement of Process-Driven Systems. *Big Data and Cognitive Computing* 7, 3 (Aug. 2023), 139. doi:10.3390/bdcc7030139

[14] Manuela Dalibor, Malte Heithoff, Judith Michael, Lukas Netz, Jérôme Pfeiffer, Bernhard Rumpe, Simon Varga, and Andreas Wortmann. 2022. Generating Customized Low-Code Development Platforms for Digital Twins. *Journal of Computer Languages* 70 (June 2022), 101117. doi:10.1016/j.cola.2022.101117

[15] Manuela Dalibor, Judith Michael, Bernhard Rumpe, Simon Varga, and Andreas Wortmann. 2020. Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits. In *Conceptual Modeling*, Gillian Dobbie, Ulrich Frank, Gerti Kappel, Stephen W. Liddle, and Heinrich C. Mayr (Eds.). Springer International Publishing, Cham, 377–387. doi:10.1007/978-3-030-62522-1_28

[16] Jürgen Dobaj, Andreas Riel, Georg Macher, and Markus Egretzberger. 2023. Towards DevOps for Cyber-Physical Systems (CPSs): Resilient Self-Adaptive Software for Sustainable Human-Centric Smart CPS Facilitated by Digital Twins. *Machines* 11, 10 (Oct. 2023), 973. doi:10.3390/machines11100973

[17] Fabrizio Fornari, Ivan Compagnucci, Massimo Callisto De Donato, Yannis Bertrand, Harry H. Beyel, Emilio Carrión, et al. 2025. Digital Twins of Business Processes: A Research Manifesto. *Internet of Things* 30 (March 2025), 101477. doi:10.1016/j.iot.2024.101477

[18] Saumya Gupta, Madhulika Bhatia, Meenakshi Memoria, and Preeti Manani. 2022. Prevalence of GitOps, DevOps in Fast CI/CD Cycles. In *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, Vol. 1. IEEE, Faridabad, India, 589–596. doi:10.1109/COM-IT-CON54601.2022.9850786

[19] Jerome Hugues, Anton Hristosov, John J. Hudak, and Joe Yankel. 2020. TwinOps - DevOps Meets Model-Based Engineering and Digital Twins for the Engineering of CPS. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (MODELS '20)*. Association for Computing Machinery, Virtual Event, Canada, 1–5. doi:10.1145/3417990.3421446

[20] István Koren, Felix Rinker, Kristof Meixner, Moritz Kröger, and Michael Zeng. 2023. Implementing DevOps Practices in CPPS Using Microservices and GitOps. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, Sinaia, Romania, 1–4. doi:10.1109/ETFA54631.2023.10275433

[21] Daniel Lehner, Sabine Wolny, Alexandra Mazak-Huemer, and Manuel Wimmer. 2020. Towards a Reference Architecture for Leveraging Model Repositories for Digital Twins. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vol. 1. IEEE, Vienna, Austria, 1077–1080. doi:10.1109/ETFA46521.2020.9212109

[22] Daniel Lehner, Jingxi Zhang, Jérôme Pfeiffer, Sabine Sint, Ann-Kathrin Splettstößer, Manuel Wimmer, and Andreas Wortmann. 2025. Model-Driven Engineering for Digital Twins: A Systematic Mapping Study. *Software and Systems Modeling* 24, 5 (March 2025), 1339–1377. doi:10.1007/s10270-025-01264-7

[23] Kalle Lyytinen, Barbara Weber, Markus C. Becker, and Brian T. Pentland. 2023. Digital Twins of Organization: Implications for Organization Design. *Journal of Organization Design* 13, 3 (Sept. 2023), 77–93. doi:10.1007/s41469-023-00151-z

[24] Jezreel Mejía, Mirna Muñoz, Álvaro Rocha, Yasmin Hernández Pérez, and Himer Avila-George (Eds.). 2024. *New Perspectives in Software Engineering* (1 ed.). Number 1135 in Studies in Computational Intelligence. Springer Nature Switzerland, Cham. doi:10.1007/978-3-031-50590-4

[25] Judith Michael, Loek Cleophas, Steffen Zschaler, Tony Clark, Benoit Combemale, Thomas Godfrey, et al. 2025. Model-Driven Engineering for Digital Twins: Opportunities and Challenges. *Systems Engineering* 28 (April 2025), 659–670. doi:10.1002/sys.21815

[26] Concetta Semeraro, Mario Lezoche, Hervé Panetto, and Michele Dassisti. 2021. Digital Twin Paradigm: A Systematic Literature Review. *Computers in Industry* 130 (Sept. 2021), 103469. doi:10.1016/j.compind.2021.103469

[27] Monique Snoeck. 2014. *Enterprise Information Systems Engineering: The MERODE Approach* (1 ed.). Springer International Publishing, Cham. doi:10.1007/978-3-319-10145-3

[28] Monique Snoeck, Charlotte Verbruggen, Johannes De Smedt, and Jochen De Weerdt. 2023. Supporting Data-Aware Processes with MERODE. *Software and Systems Modeling* 22, 6 (March 2023), 1779–1802. doi:10.1007/s10270-023-01095-4

[29] Prasad Talasila, Cláudio Gomes, Peter Høgh Mikkelsen, Santiago Gil Arboleda, Eduard Kamburjan, and Peter Gorm Larsen. 2023. Digital Twin as a Service (DTaaS): A Platform for Digital Twin Developers and Users. In *2023 IEEE Smart World Congress (SWC)*. IEEE, New York, NY, USA, 1–8. doi:10.1109/SWC57546.2023.10448890

[30] Wil van der Aalst. 2021. Concurrency and Objects Matter! Disentangling the Fabric of Real Operational Processes to Create Digital Twins. In *Theoretical Aspects of Computing – ICTAC 2021*, Antonio Cerone and Peter Csaba Ölveczky (Eds.). Vol. 12819. Springer International Publishing, Cham, 3–17. doi:10.1007/978-3-030-85315-0_1