Model-based Digital Twin Engineering: Insights, Challenges, and Future Directions

Philipp Zech[1*†], Souvik Barat[2†], Benjamin Nast[3†], Bentley Oakes[4†], Judith Michael[5†], Steffen Zschaler[6†], Balbir Barn[7], Ruth Breu[1]

[1]Department of Computer Science, University of Innsbruck, Austria.
[2]TCS Research, Tata Consultancy Services, India.
[3]Institute of Informatics, University of Rostock, Germany.
[4]Department of Computer and Software Engineering, Polytechnique Montréal, Canada.
[5]Programming and Software Engineering, University of Regensburg, Germany.
[6]Department of Informatics, King's College London, United Kingdom.
[7]Faculty of Science and Technology, Middlesex University, England.

*Corresponding author(s). E-mail(s): philipp.zech@uibk.ac.at;
Contributing authors: souvik.barat@tcs.com; benjamin.nast@uni-rostock.de; bentley.oakes@polymtl.ca; judith.michael@ur.de; szschaler@acm.org; b.barn@mdx.ac.uk; ruth.breu@uibk.ac.at;
†These authors contributed equally to this work.

Abstract

This article presents a systematic literature survey on model-based digital twin engineering (MBDTE). We introduce a novel taxonomy for categorizing MBDTE approaches and provide definitions of both MBDTE and the models it employs. Model-based engineering (MBE) leverages models as essential pillars of the development process, enabling teams to clarify requirements, streamline design, specify behavior, and perform rigorous verification and validation across the entire system life cycle. Digital twins (DTs) are software systems that mirror cyber-physical, socio-economic, or biological entities, systems, or processes. Built from models and data, DTs support high-impact applications including planning, monitoring, control, and optimization of their physical counterparts. The model-centric nature of DTs has naturally sparked exploration into harnessing MBE for DT engineering and operation. However, this exploration for now has created a fragmented landscape of partial solutions. To address this challenge, our survey analyzes 47 peer-reviewed publications across four dimensions, viz., model characteristics, data integration, implementation technologies, and empirical evidence, to map the current state of practice, identify critical research gaps, and avenues for further exploration.

Keywords: Model-based Engineering, Digital Twins, Model-based Digital Twin Engineering, Taxonomy

1 Introduction

The ever-increasing complexity and connectivity of modern systems-of-systems creates pressure for increased speed of development, reduced costs, and improved safety and sustainability. Digital twins (DTs) describe an emerging information system construct that has the potential to tackle these growing concerns. DTs refer to virtual representations of physical or cyber-physical objects, systems, or processes [59]. Recently, there has been a notable increase in attention and utilization of DTs across various industries [60] such as manufacturing [61], healthcare [62], urban planning [63], and others [64–70]. The primary concept underlying a DT is to generate a comprehensive and adaptable virtual (digital) representation of a tangible entity in the real world [71]. This facilitates enhanced comprehension, observation, and control of the entity's behavior [72] as outlined in Fig. 1.

1.1 The Digital Twin Concept

Digital twins (DTs) are developed through the integration of models and data derived from diverse sources, such as sensors or other Internet of things (IoT) devices. This integration aims to construct a dynamic and precise digital representation of the corresponding real system with the digital representation undergoing regular updates to accurately reflect real-world changes. DTs provide a variety of advantages, including the ability to monitor in real-time, simulate and analyze data, utilize predictive analytics, enable remote control and automation, and facilitate life cycle management [61, 73, 74].

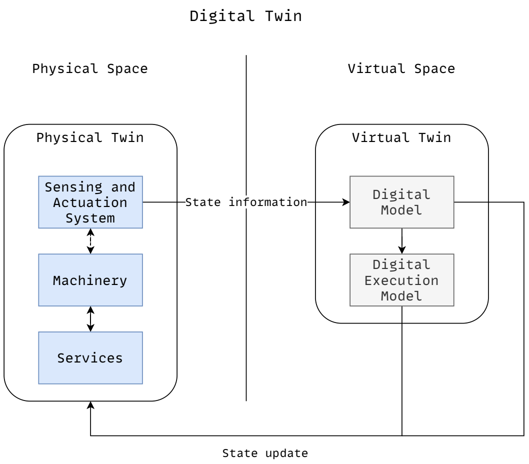


Fig. 1: Concept of a behavior-centric DT after Stary et al. [72] where a *digital execution model* controls the behavior of a physical twin.

A DT consists of (i) models, (ii) data, and (iii) a tangible entity existing in the real world. According to ISO/IEC 30173:2023 [75], DTs "enable convergence between the physical and digital states at an appropriate rate of synchronization". Tao et al. [76] define a five dimensional model comprising (i) a physical entity, (ii) a virtual entity, (iii) services, (iv) data, and (v) connections as depicted in Fig. 2. Crucially, Tao et al.'s model establishes a DT around a common data model which allows tailoring the DT towards the requirements posed by its use cases and applications.

1.2 Engineering Digital Twins

Typically, Digital twin engineering (DTE) encompasses various stages and factors, which vary depending on the intricacy and characteristics of the real world entity being replicated in the virtual realm [77]. Usually, the initial step establishes a clear set of objectives and determines the scope of the DT. This process inherently involves identifying the specific data types and sources that will be utilized by the DT. The subsequent phase typically focuses on model development and management. This entails determining the specific models that are constructed, the manner in which they are constructed, and how they are maintained throughout the lifespan of the real-world entity. This includes updating the models in accordance with changes in the real-world entity.

Given the available models and data, the *coupling* of these two, i.e., the combination of models with data, is the subsequent and most crucial

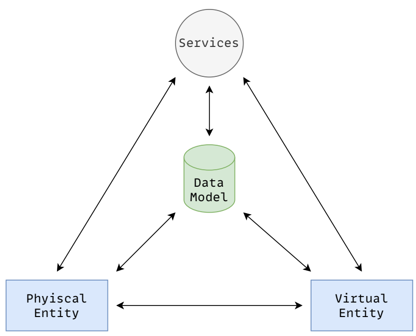


Fig. 2: Five-dimensional DT model, Tao et al. [76].

stage in DTE. Here, *coupling* describes the successful resolution of the challenge of integrating models and data [78], which is vital for the dynamic and accurate virtual representation of the real world entity by the DT.

During this process of engineering the DT, according to Bordeleau et al. [79], three main challenges are to be overcome, viz.

1. the use and integration of a set of heterogeneous models that are required to address the different aspects and disciplines of a system,
2. the synchronization of the DT with runtime data, and
3. the co-development and management of the evolution of DTs by teams of engineers.

Aligning with Bordeleau et al., we also support the notion that Model-based engineering (MBE) plays a crucial role in tackling these challenges and, on a broader scale, in the development and management of DTs. Specifically, MBE is beneficial for managing complexity, improving automation, and allows for the definition of domain- and platform-specific models [80]. In recent decades, MBE has experienced notable successes in domains such as railway systems, automotive industry, business process engineering, and embedded systems [81]. These successes have led to the extensive adoption of MBE, where models are crucial components in contemporary engineering processes [82]. Therefore, we wish to explore the utilization of MBE in engineering DTs, cf. Model-based digital twin engineering (MBDTE).

1.3 Research Questions and Contributions

To frame a systematic review that outlines the state of MBDTE, we followed the step-wise process laid out by Kitchenham and Charters for systematic literature reviews in software engineering [83]. To this end, we initially articulated a single, concrete motivation: the absence of a comprehensive surveying the field, which investigates the use of MBE for DTE that goes beyond cataloging modeling tools and code generation practices (cf. Lehner et al [84], Sec. 2.4). This gap set the overall goal — to map current practice across the full engineering life cycle — and in addition ensured that every subsequent decision was aligned with this practical objective.

Next, we formalized our information need with PICOC [85]. The *Population* was the body of peer-reviewed DT studies; the *Intervention* any engineering method or tool they employed; there was no formal *Comparison* because the review is exploratory; the intended *Outcomes* were the dimensions in our taxonomy (Model, Data, Implementation, Evidence); and the *Context* allowed any application domain.

With goal and PICOC fixed, two authors independently brainstormed candidate questions. Multiple drafts emerged, which we piloted on a seed set of randomly selected papers. This pilot served as an early feasibility filter: if the literature did not allow to answer a research question, it was dropped. This resulted in the seven central research questions (cf. Tbl. 1).

We then iterated the wording for precision and non-overlap, replacing vague verbs such as (e.g., "deal with") by observable actions (e.g., "specify", "verify", or "integrate"), and refined any question that tried to cover two or more facets at once into dedicated subquestions. Finally, echoing Kitchenham's call for bias control [86, 87], the provisional list of research questions was validated by three authors that were not involved in the process of establishing the research questions. Specifically, they judged each question for clarity, relevance, and feasibility. Their suggestions prompted only minor tweaks. After this validation, the questions were frozen before the full search and extraction began. Following Kitchenham and Charters [83] we finally arrived at a concise set of seven research questions that are both traceable to the survey's objectives and answerable with the evidence reported in the primary studies, minimizing bias and the risk of empty results.

The evaluation conducted in this study presents an analysis of the current state of MBDTE and serves as a foundation for addressing the research questions outlined in Tbl. 1. Our evaluation helps to identify unresolved research challenges in MBDTE. Specifically, RQ1-5 aim at outlining a comprehensive picture of the current state of application of MBE for DTE. A strong emphasis is put on how this application is done practically and also how models and data are combined together. The goal of RQ6 and RQ7 is to draw an outlook on potential future research areas, not only in MBDTE but also w.r.t. novel challenges posed to MBE in being applied for

DTE. Finally, in the spirit of Paige and Cabot [88] and the reporting framework of Gil et al. [89], we delineate characteristics of MBDTE that should be reported in scientific contributions to enhance their value.

The evaluation criteria employed in our study are specifically designed for the assessment of MBDTE, taking into account the unique characteristics of DTE in conjunction with MBE. The established criteria were systematically applied by us to categorize existing approaches in the MBDTE literature. We conducted a rigorous literature review to search for and select publications related to MBDTE, resulting in a total of 47 papers. For each publication, we were able to evaluate the introduced MBDTE approach uniquely according to the defined criteria. This finding strongly suggests the adequacy of the evaluation criteria itself.

We believe that this article is beneficial to both the DT engineering and larger modeling communities. It uses common terminology to provide a comprehensive state-of-the-art review of the field and proposes research challenges and recommendations to provide a firm foundation for the active MBDTE research area.

Organization Sec. 2 presents relevant background on MBE, DTE, and MBDTE, and discusses related work. Sec. 3 defines the evaluation criteria for MBDTE approaches. In Sec. 4, relevant publications are systematically selected and evaluated according to the criteria defined in Sec. 3. Sec. 5 presents our results and answers our research questions from Table 1. We discuss our results and position them with related work, and identify open research challenges in MBDTE in Sec. 6 before concluding in Sec. 7.

2 Model-based Digital Twin Engineering

MBDTE, compared to MBE in general, is a young research area that emerged substantially in recent years due to the generally growing demand in DT-based solutions [90, 91]. MBDTE capitalizes on the integration of prominent theoretical ideas imported from the fields of MBE (cf. Sec. 2.1) and DTE (cf. Sec. 2.2).

2.1 Model-based Engineering

MBE advocates for elevating models, cf. digital models of a systems' structure and behavior, to first-class citizens within the engineering process. We use the term *digital model* in a broad sense to include any kind of model that relies on a formal meta-model or grammar, thereby referring to formal structural, behavioral, or ontological models with well-defined semantics. MBE relies on models for the activities encompassed within engineering processes, such as requirements engineering, architectural design, behavior specification, and verification and validation [92, 93]. MBE is particularly valuable when developing complex systems where multiple stakeholders need to collaborate, requirements must be rigorously traced, and system behavior needs to be verified before implementation [94, 95]. MBE excels in domains such as systems engineering, automotive, aerospace, and cyber-physical systems, where the cost of design errors is high and where models can serve as precise communication artifacts across disciplinary boundaries [96]. While MBE introduces overhead in terms of tool adoption and modeling effort, this investment is justified when system complexity, safety requirements, or regulatory constraints demand systematic traceability and validation throughout the development life cycle [97].

The Unified Modeling Language (UML) [98] is a widely used modeling formalism in the field of software engineering. UML encompasses structural diagrams such as class or component diagrams, and behavioral diagrams like sequence or activity diagrams. The System Modeling Language (SysML) [99] is the predominant modeling formalism utilized in systems engineering. The Business Process Modeling Notation (BPMN) is the established norm for representing processes visually for the purpose of business analysis in process modeling [100]. The FTG+PM framework, unrelated to systems or business processes, allows for explicit modeling of model transformations between defined models using different formalisms [101]. In recent decades, specialized modeling languages for enterprise modeling, such as ArchiMate [102, 103] and Extended Enterprise Modeling Language [104] have been created. However, modeling languages like UML and SysML are often viewed as overly sophisticated

Table 1: Research questions investigated in our survey.

RQ1: What is the current state of MBE for DTE?
RQ1.1: What is the purpose of using MBE in DTE?
RQ1.2: How widespread—in terms of domains—is the use of MBE in DTE?
RQ1.3: Does the use of MBE improve DTE and how?
RQ1.4: What is the level of maturity of MBE in DTE?
RQ1.5: In which domains is MBE most frequently applied for DTE?
RQ2: How is MBE applied for DTE?
RQ2.1: What MBE concepts are used in DTE?
RQ2.2: What modeling formalism and paradigms are used in developing DTs?
RQ2.3: What is the level of abstraction of models?
RQ2.4: Are models characterized by static or dynamic properties, and do they exhibit stochastic or deterministic behavior?
RQ2.5: What do models in a DT represent: the physical twin, the computational environment/infrastructure, or both?
RQ2.6: What types of systems are modeled in which way, i.e., what types of models are used for what purpose?
RQ2.7: How is model correctness assured?
RQ2.8: How are models developed and managed?
RQ2.9: How are models combined?
RQ3: How are models and data coupled?
RQ3.1: What sources of data and what types of data are utilized?
RQ3.2: Does data exhibit a uni- or multi-modal nature?
RQ3.3: How often are models and data synchronized?
RQ3.4: How are models and data synchronized?
RQ3.5: Do models only consume data or also emit data, e.g., control values?
RQ4: How is MBE implemented in DTE, specifically what are the frameworks, technologies, development languages, and modeling tools are used?
RQ5: In which life cycle phases of a DT is research located?
RQ6: What new MBE research challenges are created by its application in a DT context?
RQ7: What aspects of MBDTE merit deeper investigation and more thorough reporting in literature?

and unsuitable for certain applications due to their adaptability and complexity [105, 106]. This perception has led to the construction of Domain-specific Languages (DSLs) [107]. The term DSL refers to modeling and programming languages tailored to a given domain or problem space. These languages are characterized by their narrow emphasis, greater expressiveness, higher-order abstractions, and constrained scope [107]. DSLs can enhance code or model productivity and comprehensibility in specific issue domains, but their construction involves careful planning and understanding of the domain's goals and limits [108].

The advantages of MBE include improved communication and enhanced system comprehension, early identification of design problems, increased reusability, more informed decision-making, automation and code generation, improved maintenance and evolution, systematic change control, and integration and interoperability [109]. The widespread acceptance of MBE can be attributed to the significant capabilities of models in capturing various aspects of systems, including (i) static aspects related to architecture, (ii) dynamic aspects related to behavior, (iii) (non-) deterministic aspects related to execution, and (iv) performance-related and quality-related aspects. MBE also excels at managing complexity, delivering automation and tailoring engineering processes to different domains with domain- and platform-specific modeling. Therefore, the utilization of MBE in engineering DTs is beneficial to develop effective solutions [110].

2.2 Digital Twin Engineering

DTs describe virtual replicas of physical objects, systems, or processes that are characterized by [76] (cf. Fig. 2, p. 2)

- a *virtual twin*, i.e., the assemblage of system, simulation, optimization, and operating models which together replicate the physical object, system, or process (cf. the real world entity),
- a *physical twin*, i.e., the physical object, system, or process,
- *interchanged data and connections*, i.e., bidirectional communication, or *twinning*, between the virtual and physical twins, respectively,
- common data, and
- application services,

and may manifest at three different levels, depending on the degree of automated interchange of data [111] as illustrated by Fig. 3.

The primary objective of DTs is to effectively integrate models and data to establish virtual twins of real-world entities. These replicas can be utilized for the purpose of simulating and enhancing production processes [61, 65, 111], or predictive maintenance. DTs have also been employed to construct virtual models of patients [62], for maintenance of dams and decision support in or before a flood event [112], for simulating urban environments [63], and in replicating power plants, wind farms, and oil rigs [64]. To comprehensively represent the various aspects of these physical twins, DTs typically utilize Multi-paradigm modeling (MPM) [113]. This involves employing a combination of diverse modeling formalisms and paradigms to accurately describe the physical entity at the desired level of abstraction using a collaborative environment for model management [114, 115].

In addition to the essential formal modeling frameworks used to describe the manifestation of DTs, data serves as the second crucial foundation of DTs [116]. In particular, the development and success of DTs heavily rely on the presence of reliable, high-quality, and continuous data, which is essential for enabling the mentioned applications. Without the automatic exchange of data, the resulting virtual replicas merely represent a *digital model* as categorized by Kritzinger et al. [111] (cf. Fig. 3).

The availability of high-fidelity models and high-quality data alongside data analysis and simulation ultimately enables the most notable characteristic of DTs — their ability to combine models and data to conceptualize the condition of the physical entity in real-time, facilitating the monitoring of performance [117]. This concept bears resemblance to the paradigm of *models@run.time* [118], which explores methods for abstracting the runtime state of a system using models. In particular, models@run.time aims to improve the flexibility, adaptability, and efficiency of software systems by employing models that are updated and utilized during the runtime of the system. These models then are used to make informed decisions and guide the runtime behavior of the system, e.g., to improve resource management [119], or self-healing and fault tolerance [120, 121]. These executable models also comprise, among others, simulation models (cf. Modelica [122]) or agent-based models [123]. By utilizing various modeling formalisms and paradigms at different levels of abstraction, such as MPM, it becomes feasible to monitor the runtime state at multiple levels of abstraction. However, the data and state information thus available can further be utilized to update models based on runtime data in order to accurately represent the physical entity. Real-time data is further utilized in behavioral models to facilitate simulations of prospective system behavior across various conditions [124]. Consequently, this enables the deduction of optimal control parameters to be transmitted to the real-world entity.

This *combination* of models and data inside the DT however is only possible by choosing the right "enablers" as argued by Oakes et al. [125]. Enablers are the internal building blocks that specify how models are processed, simulated, and visualized in the context of the DT. Following Tao et al.'s model (cf. Fig. 2), the enablers are computational components that exist between models, data, and services, thus utilizing models and data to enable the services provided by the DT [125]. The selection of enablers thus substantially affects the DT's usability and success, and their details in terms of computation and formalisms are crucial for DTE.

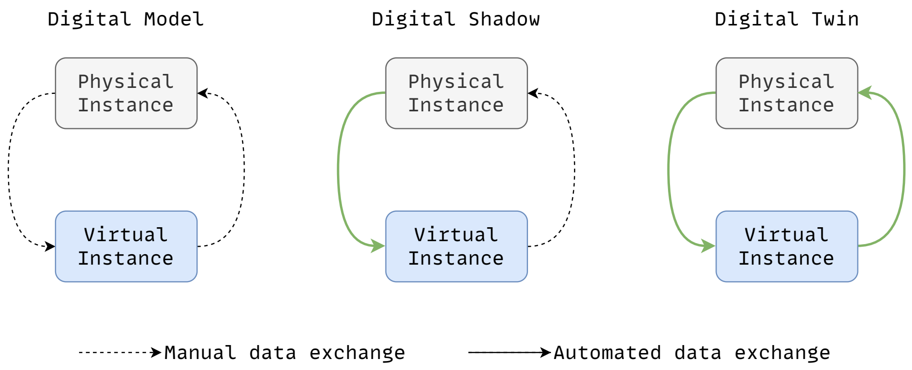


Fig. 3: Types of DTs given the automation of data exchange [111].

2.3 Model-based Digital Twin Engineering

After our discussion of MBE and DTE, we now introduce our notion of MBDTE as it emerged out of the combination of these two fields.

2.3.1 Models in MBDTE

Up to this point, we have used the term *model* rather broadly in the sense of Stachowiak [126]. He postulates three important characteristics for models: the mapping feature (it represents an original), the reduction feature (it represents not all but only some properties of the original), and the pragmatics feature (the purpose a model is used for when replacing the original). In addition to these characteristics, the models we can use in DTs need formalized constructs for their representation, i.e., a modeling language also referred to as *modeling formalism* (cf. Mayr and Thalheim [127] for further model characteristics). Our broad understanding can be attributed to the polymorphic nature of DTs where models are used for describing the structure of a DT or its environment, its behavior, and its data. This usage aligns with our initial validation study (cf. Sec. 4.2), i.e., the underlying models of a DT manifest in formats such as system models, physical models, simulation models, process models, but also statistical or machine learning (cf. numerical models).

In our understanding of what a model in MBDTE comprises, we follow Combemale et al. [128] who introduce three types of models in their *Models and Data* (MODA) framework and extend it by one additional type, viz. *socio-economic models* [129, 130]:

- *engineering models* pertaining to the definition and representation of a specific system [131],
- *scientific models* that function as a representation of certain aspects of a phenomenon in the natural world [132], and
- *machine learning models* that are generated by autonomous learning algorithms from sample data (cf. training data) to make predictions or decisions without being explicitly programmed to handle a task, and
- *socio-economic models* that include a mix of models of human behaviors and scientific models that represent behaviors emanating from social science theory base.

Following this categorization, we consider models relevant for our study to emerge from the group of *engineering models*, i.e., models used to develop and evolve a system for a specific purpose, ideally with support for automation [128].

Our study's understanding of "model" substantially coincides with Boyes and Watson's dimension of physical entity models [77], including *economic*, *environmental*, *functional*, *physics*, *safety*, *security*, and *spatial* models, with the constraints mentioned above. Following Boyes and Watson, this then also includes, e.g., models of non-engineered systems (cf. biological models as subcategory of environment models, process models as a subcategory of functional models, or organizational models as part of economic models). . In that sense, models are used for aggregating and managing disparate information about a

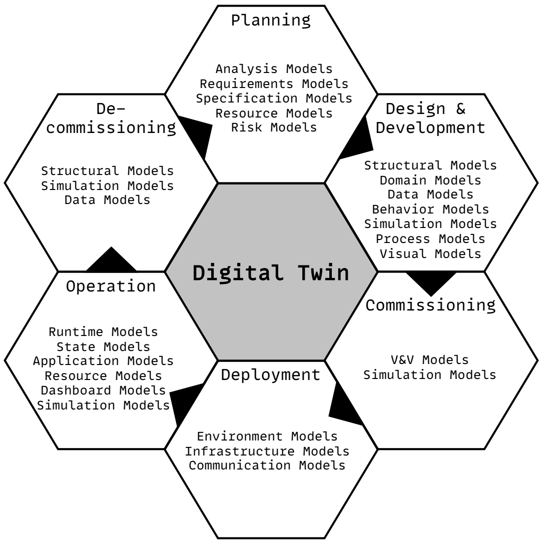


Fig. 4: Model types used along the life cycle.

real-world entity, serving as a technical baseline regarding a real-world entity (cf. Model-based Systems Engineering [80]). However, it is important to note that models are inherently approximations, and many DTs employ multiple models to achieve better fidelity.

Broadly spoken, DTs can use any sort of model that provides a sufficiently accurate and understandable representation of the real world entity along its life cycle [133], and should be:

-
- sufficiently grounded in first principles (e.g., physics-based modeling, system modeling) to provide a sound and systematic basis for calibration and updating with operational data
- sufficiently accurate and detailed that updates will be useful for the application of interest, and
- sufficiently analyzable and executable such that they allow to make decisions about the application within a reasonable time.

Fig. 4 provides an overview of the different types of models as employed along the life cycle of the physical entity. DTs may accompany the physical twin across multiple life cycle phases, though in practice most implementations focus on specific stages such as operation and maintenance. The conventional life cycle comprises six phases [134]:

- *Planning* Planning involves setting goals, allocating resources, assessing risks, and setting timelines, budgets, and communication techniques. Additionally, it involves incorporating stakeholders, creating contingency plans, and documenting the process to ensure project success. This stage aims to review and organize project parts before implementation.
- *Design* Concepts and requirements from the planning phase are turned into detailed specifications and drawings during the design phase. Prototypes, mock-ups, or simulations are often used to ensure the final product or solution meets objectives and requirements. Iterative improvement, prompted by feedback and testing, is usual in this stage to achieve the most optimal and streamlined design.
- *Commissioning* During commissioning, the DT undergoes thorough testing, verification, and validation to guarantee compliance with specifications and proper operation in its intended environment (cf. acceptance testing [135]). Upon validation, the system is given to the operator or end-users, along with training and documentation for smooth operation.
- *Deployment* The DT is operationalized during the deployment process, including installation, setup, and integration with existing infrastructure. Testing here ensures project objectives are met or modifications are made before full operation (cf. integration or smoke testing [136]).
- *Operation* During the operating phase, the DT is used and maintained according to established criteria. Monitoring is ongoing to ensure optimal performance, stability, and functionality. This stage involves routine maintenance, problem-solving, and updates to improve efficiency and address any issues identified during the project's life cycle.
- *De-commissioning* In the decommissioning phase, the DT is meticulously shut down and removed from service. This typically involves discontinuing operations, repurposing assets, and closing related facilities or infrastructure. This phase includes data transfer, archiving, and transition methods for a smooth departure from the working environment.

When talking about the real entity's life cycle phases in which a DT eventually is employed, we should be mindful that DTs are not restricted to a single domain but various domains, all with their established engineering life cycle and accompanying phases. In an effort to provide a comprehensive overview, we thus decided to stick with the above decomposition of life cycle phases following Cao et al. [134]. For example, the separate design and commissioning phases may be superficial in the field of systems engineering, however, they are of utmost relevance in construction engineering, where testing can not be done continuously but only once a building is complete. In addition, along the DT's life cycle we have to distinguish between the models of the DT and models in the DT, e.g., those that describe the DT (e.g., structural or state models) and those that operate in the DT (e.g., simulation or runtime models) [110].

2.3.2 A Definition of MBDTE

In accordance with our discussion so far, we define models in MBDTE as follows:

Definition 1. *A model is a representation of a system, process or other physical entity (e.g., biological or organizational systems) of interest, approached from a specific viewpoint (e.g., a specific perspective on a system's aspects, like structure or behavior) and encompasses characteristics serving a particular objective. A model is inherently an abstraction selectively emphasizing relevant properties while disregarding irrelevant properties, at a predetermined level of precision, and is formulated using a specific modeling language that defines the models's syntax and semantics.*

In their survey on DT models, Tao et al. [137] provide valuable insight into the various characteristics models of a DT express. By conducting a *multi-perspective* and a *multi-aspect* analysis of DT models, they investigate characteristics like

- the application domain of DT models,
- the hierarchy of DT models,
- the disciplinary of DT models,
- the dimensionality of DT models,
- the universality DT models,
- the functionality of DT models,

for the former, and characteristics like

- model construction,
- model assembly,
- model fusion,
- model verification,
- model modification, and
- model management

for the latter. This comprehensive list of multi-perspective and multi-aspect characteristics is essential for the evaluation criteria introduced in Sec. 3 as it is more comprehensive than the characteristics of models usually treated in the context of MBE (cf., Felderer et al. [138]). Specifically, model assembly and fusion as well as model modification (cf., models@run.time) are paramountly relevant in the context of DTs but rather irrelevant, e.g., in model-based testing. Consequently, our upcoming evaluation criteria will reflect these aspects (among other, earlier discussed aspects of MBE and DTE). In that respect, we are not extending Tao's characteristics but instead integrate these characteristics into our evaluation criteria alongside criteria motivated from MBE and DTE (cf. Secs. 2.1 and 2.2).

For the purpose of our study, we thus define MBDTE in the following way:

Definition 2. *MBDTE is the structured utilization of modeling techniques to facilitate various activities related to system requirements elicitation, design, analysis, verification and validation, deployment, and operation and maintenance of a DT and its real world entity. MBDTE can be initiated at any life cycle stage, though it is most commonly applied beginning at design & development. When initiated early, MBDTE may be sustained throughout subsequent phases including operation and maintenance, though partial life cycle coverage is common in practice. MBDTE involves close collaboration, and model and data exchange with other often complementary disciplines and the integration of reliable system data in digital format from these disciplines within a DT.*

MBDTE is the successful application of techniques and practices from MBE for the purpose of (i) engineering and (ii) operating DTs thereby capitalizing on MBE's prime qualities, viz. managing complexity, high support for automation, and domain- and platform-specific modeling. That said, MBE serves a dual duty in DTE. On the one side, MBE is applied for the actual engineering of the DT itself, e.g., structural and behavioral modeling, and subsequent model transformations to, for example, generate runtime artifacts like a data model from the DT's structural and behavioral specification. At runtime, e.g., when operating the

DT, this data model then, in conjunction with the behavioral models, is employed for model-based monitoring and control of the DT by leveraging concepts from models@run.time. MBDTE thereby further adopts key theoretical concepts from various disciplines, viz. MBE, IoT, big data, Artificial intelligence (AI) and data analytics for establishing coherent connections between the physical and digital realms. Whereas the state of the art of IoT, big data, AI and data analytics has already been investigated [139–141], the application of MBE for DTE currently lacks such a systematic treatment beyond cataloging modeling tools and code generation practices (cf. Lehner et al [84]). This is further elaborated on below in our related work section where aside from discussing studies directly relevant to the specific topic of our work, viz. the application of MBE for DTE, we also give an overview of recent meta reviews on DT. This allows us to draw a broader picture on DTs, but they are topically less relevant for our work as they do not touch on MBE and related topics.

2.4 Related Work

A wide range of domain-specific surveys have contributed to mapping the conceptual and technical landscape of DTs. In manufacturing and industrial systems, recent reviews such as Semeraro et al. [142] and Baricelli et al. [143] provide a solid foundation, while others explore DT applications in intelligent transportation [144], construction and built environments [145], healthcare [146], and smart energy systems [147]. These surveys collectively reflect the growing maturity of the field and highlight recurring dimensions such as fidelity, synchronization, life cycle integration, and semantic interoperability. In parallel, a small number of recent broader reviews aims to synthesize cross-domain findings and investigate DTs at a meta level. Among them, Bühler et al. [148] offer an umbrella review of 26 earlier survey papers, identifying recurring topics across various sectors. Complementing this, Zech et al. [60] present an exploratory questionnaire aiming at grounding recurring practices, barriers, and desiderata in DTE which in turn were identified by a meta survey. Collectively, these studies provide a thorough overview on DTE. Admittedly, there exist many surveys on DTs, such that covering all those is not the purpose of this study (and also not feasible). Hence, given the focus of our work however, viz. classifying MBDTE approaches, we thus now conclude this section with a concise discussion of related survey-based classification studies that also touch on MBE-related topics while establishing evaluation criteria similar to ours.

Dalibor et al. [149] provide a cross-domain mapping study on digital twins (DTs) in software engineering, focusing on six questions: DT purposes and users (RQ-1), conceptual properties (RQ-2), and the processes of engineering, deploying, operating, and evaluating DTs (RQ-3-6). Their findings, especially for RQ-3-6, align with our focus on DTE but span multiple domains rather than our MBDTE approach. While Dalibor et al. discuss models primarily in terms of physical constraints and appearances, categorizing them as "simulations, physical models, and geometric models," their results on DT engineering dimensions complement our work.

In contrast, Gil et al. [150] assess open-source DT frameworks using criteria based on ISO 23247:2021 [151], evaluating fourteen frameworks and a case study to explore applications, limitations, and model-simulation capabilities. Oakes et al. later propose a DT reporting framework with 21 characteristics on the capabilities, architecture, and evolution of a DT to ensure that they are systematically reported in DT experience reports, which often lack detail [89, 152]. Characteristics include the *system-under-study*, *DT services*, *life cycle stages*, the *DT constellation*, and the *DT twinning process*.

Autiosalo et al. [153] propose FDTF, a framework for defining DT characteristics in industrial contexts, categorizing DT implementations, and guiding future applications. Newrzella et al. [154] suggest a five-dimensional model for DT applications that highlights both the physical and virtual entities and user-specific outcomes. Meanwhile, Uhlenkamp et al. [155] introduce a DT maturity model for Industry 4.0, assessing DTs across their life cycle and identifying potential for enhancement in new scenarios.

Zambrano et al. [156] present a modular DT platform aimed at supporting component reuse in DT applications, categorizing models by metadata such as fidelity and model type. Lastly, Hakiri et al. [157] survey DTs within IoT, categorizing DT models into physics-based, analytics-based, visual, and DES models, and identifying ten standardization efforts, software frameworks, and related research challenges.

Recently, Lehner et al. presented a systematic mapping study on the application of Model-driven engineering (MDE) techniques to DTs [84]. Their study provides a broad overview by categorizing existing research based on MDE techniques such as model-to-model transformation, code generation, and model interpretation, applied across various domains. While their work offers a comprehensive mapping of current research, it lacks the in-depth analysis and critical insights into chosen approaches, which our work addresses by providing a detailed evaluation framework to explore MBDTE for more thorough investigation.

In synopsis, related works consider models in DTs, but often do not decompose them beyond assigning them a type. In contrast, we wish to deeply investigate the use of models in DTs including the model engineering process itself. Notably, Lehner et al. [84] primarily investigate the use and application of models. In contrast, we develop a framework for in-depth investigation and consider our contribution orthogonal to Lehner et al.

3 Evaluation Criteria

This section outlines the evaluation criteria used to systematically categorize approaches related to MBDTE. The objective of these evaluation criteria is to analyze and evaluate current and prospective methodologies and trends in the field of MBDTE with a focus on available evidence. We propose four dimensions: *model*, *data*, *implementation*, and *evidence*, where each dimension defines multiple criterion groups along which we assess current approaches in MBDTE. Criteria are open-valued, as a pre-compilation of possible values is not feasible due to the fields's immaturity. Instead, we aim at capturing typical manifestations with our evaluation criteria.

3.1 Criteria Development

In defining our evaluation criteria for MBDTE, we adhered to the approach suggested by Usman et al. [158] for constructing taxonomies. Usman et al. propose a systematic approach consisting of four phases for constructing taxonomies in software engineering. These four phases are [158]:

1. **Planning** which comprises activities for defining the taxonomy's context and initial setting, i.e., software engineering knowledge area, objective, subject matter, classification structure type, classification procedure type and sources of information.
2. **Identification and extraction** which comprises activities for extracting and controlling the terms associated with the taxonomy.
3. **Design and construction** which comprises activities for supporting the identification and description of the dimensions, and categories and relationships; in addition, guidelines for using and evolving a taxonomy are provided.
4. **Validation** for ensuring that the selected subject matter is clearly, concisely and thoroughly classified.

After laying the context of our taxonomy (cf. Sec. 2) during initial planning, continuing with phase 2, we established an initial set of criteria along which to classify MBDTE approaches. Given these, in the third phase we then identified criteria dimensions and constructed our initial taxonomy by clustering the criteria along the identified dimensions. Finally, this taxonomy then was subjected to validation to assess its internal validity (cf. phase 4, Usman et al. [158]). To this end, we randomly selected papers during our search and selection procedure (cf. Sec. 4), specifically among those that were still included after screening abstracts to avoid using publications that do not fit the MBDTE context and setting. We iterated the following three-step protocol twice to validate our taxonomy:

1. identify proposed criteria in selected papers; if not possible, propose criteria revision and alignment;
2. identify proposed dimensions in selected papers; if not possible, propose dimension revision and alignment;
3. discuss and validate the criteria- and dimensions-to-amend in a plenum meeting to assure a common understanding of the final criteria and dimensions.

After the second iteration, no more modifications were suggested and a collective agreement was reached on the modified criteria. We will report in Sec. 4.2 which papers have been selected for this *validation* study and which modifications were triggered in our proposed criteria.

To conclude, our classification framework and its comprising evaluation criteria were obtained through a systematically conducted process [158], and subsequently were successfully applied to all 47 publications selected, which strongly suggests that it provides a general facility for evaluating and comparing approaches for MBDTE. In the following, we now move to the discussion of our evaluation criteria of our final taxonomy as applied later on for classifying and evaluating research in MBDTE (cf. Sec. 4.3).

3.2 Criteria Discussion

We have identified four groups of criteria relevant for analyzing approaches related to MBDTE. Tbl. 2 presents the complete taxonomy, structured to provide a multi-faceted view of each MBDTE approach. The table is organized into four columns: *Topic* identifies one of the four primary dimensions, which are further refined by sub-groups; *Criteria* lists the specific characteristic being evaluated; *Description* provides a concise definition of that criterion; and the final column offers a *List of Example Characteristics* to illustrate the types of values encountered in the literature for each dimension's sub-groups' criteria. Together, these elements form a comprehensive framework for classification. The four primary dimensions are:

- Model criteria (cf. Sec. 3.2.1) comprise the *specification*, *characteristics*, and *development* of models. Together, these assess the application of MBE in MBDTE to answer RQ1 and RQ2.
- Data criteria (cf. Sec. 3.2.2) assess the *specification* and *integration* of data, describing how the data aspect of DTs is incorporated in an MBDTE approach to answer RQ3.
- Implementation criteria (cf. Sec. 3.2.3) explore the *frameworks*, *technologies*, *languages*, and *tools* employed in MBDTE to provide valuable insight into the technological landscape and answer RQ4.
- Evidence criteria (cf. Sec. 3.2.4) evaluate *maturity*, *evidence measures* and the *application/use case* of a MBDTE approach under study. Collectively, evidence criteria assess the industrial applicability and utility of MBDTE as well as the actual state in research and answer RQ1 and RQ5.

Running Example

To illustrate our subsequent discussion of the proposed evaluation criteria, we now first introduce a short running example for explaining the rationale behind our decisions.

Our example is the *incubator case study* [159]. Feng et al. present a DT for an incubator designed to produce tempeh, a traditional fermented soybean product. The incubator is a simple insulated box equipped with a heating element, a fan, and temperature sensors. All components interact under a bang–bang control strategy to maintain an internal temperature of around 37.5 °C. Variations in ambient temperature or user interactions, such as opening the lid, can disrupt this system, thereby affecting the fermentation process. Fig. 5 provides both a system overview (cf. Fig. 5a) and a block diagram (cf. Fig. 5b) of this incubator.

The DT of the incubator is constructed around virtual models, such as a *plant model* expressed through differential equations that capture the thermal dynamics of the system. This plant model is continuously calibrated against real sensor data to ensure that its simulated behavior closely mirrors that of the physical incubator. The DT also has a state machine as a *controller model*, algebric equations for a *room temperature model*, and a hybrid automata for the *coupled plant + controller + room model*. Consequently, the DT not only enables real-time monitoring and visualization of historical temperature data but also supports decision-making through what-if simulations and anomaly detection. These services, offered within a service-oriented architecture, highlight how aspects such as model fidelity, data integration, and service management directly inform the evaluation criteria in our framework.

For further details on the incubator and its DT, we point the reader to numerous works discussing it [89, 159–161].

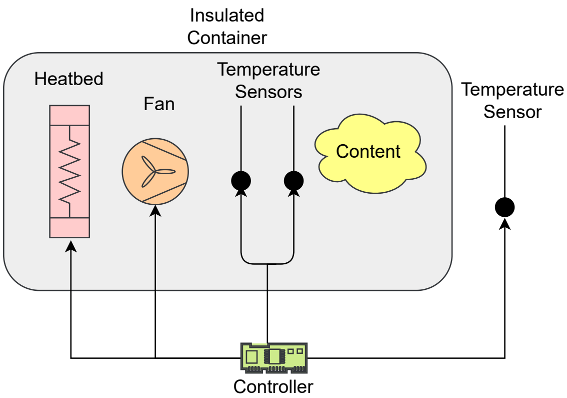
3.2.1 Model

Model evaluation criteria provide means for investigating the application of MBE for MBDTE. These criteria formally define the modeling objectives and describe what is modeled in which way:

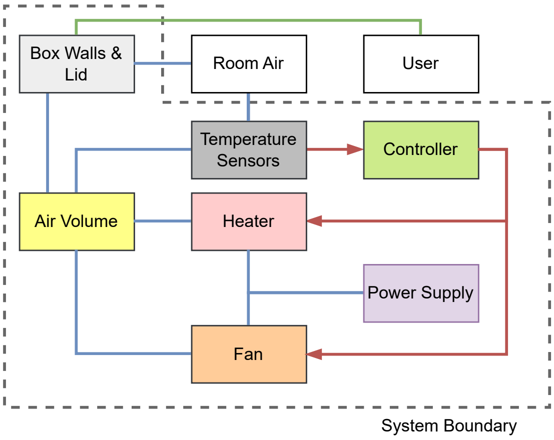
Table 2: Overview of evaluation criteria of our taxonomy for MBDTE. Best viewed on a computer screen.

Topic	Criteria	Description	List of Example Characteristics
Model Specification	Paradigm	Aspect of the physical entity in focus	Structure, behavior, function, physics, rules, ...
	Formalism	Used modeling formalisms/languages	UML, SysML, Simulink, DSLs, ...
	Universality	How general are the models	General purpose or domain-specific
	Level	Level of abstraction	Conceptual, implementation, execution, ...
	Probabilistic	Predictability of outcomes given initial conditions	Deterministic or stochastic
	Time progress	Models' mechanism for advancing time	Discrete, continuous
	Time horizon	Models' temporal scope and its capacity for representing change	Static (point in time) or dynamic (state changes)
Model Contents	Subject	The contextual role of the model within the whole ecosystem	Describing the DT, runtime environment, the process, ...
	Scope	What is modeled, the focus	CPS, socio-economic or socio-technical system, natural system, ...
	Verification	Are the models verified and validated	Syntactic, functional, or semantic correctness
	Purpose	What is to be achieved with the model(s)	Engineering or operating the DT, monitoring, control, prediction of the physical twin
	Fidelity	Degree of accuracy and resolution reflecting the physical twin	Number and accuracy of parameters, level of abstraction
Model Development	Construction	Way of creating models	Manual, automatically from data, semi-automatic, or manually from domain knowledge, ...
	Assembly	If models are hierarchically decomposed, how are they unified	Connecting mechanisms in modeling languages (i.e., associations, links), generation, automatic, manual, ...
	Fusion	If models are on the same hierarchy, how are they connected	Connecting mechanisms in modeling languages (i.e., imports, links), megamodels, manual connections, ...
	Management	Storage of models	Repository, knowledge base, ...
	Modification	Change/update mechanism of models	Manually, semi-automatic, or completely autonomous
Data Specification	Data Source	Origin of the data	Sensors, web services, standards, ...
	Data Frequency	Acquisition interval	Per second, minute, hour, ...
	Data Type	Value domain of the possible values	Real-valued, binary, categorical, ...
	Modality	How many distinct modes of information are involved	Unimodal or multi-modal
Data Integration	Direction	Orientation of information flow	Data flow into models or data created from models
	Coupling	Connection of data and models	Linked, attached, annotated, ...
Implementation	Framework	Used frameworks	Eclipse BaSyx, Eclipse Hono, ...
	Technology	Used technologies	MQTT, OPC UA, InfluxDB, ...
	Language	Programming languages	Python, Java, C#, ...
Evidence	Maturity	Readiness level of the DT	Conceptual study, lab prototype , used in production,...
	Measures	Qualitative or quantitative assessment criteria	Walk through, case study, testing, ...
	Application	Concrete application of the DT, what is it used for	Domain dependent, e.g., identify logistics bottlenecks, improve air quality, or predict maintenance, ...
	Phase	Life cycle phase of the physical twin	Planning, design & development, commissioning, deployment, operation, de-commissioning

which aspects of the real world entity are captured, how they are captured, and at which level of abstraction. In addition, they investigate model development as well as application specific characteristics of the models. In the following, the criteria groups *specification*, *model contents* and *development* are described in more detail.



(a) System diagram of the incubator.



(b) Block diagram of the incubator.

Fig. 5: System (cf. Fig. 5a) and block (cf. Fig. 5b) diagram of the incubator system [160].

Specification

The criteria in this group target the specification of models in terms of their formal and theoretical underpinnings as well as their manifestation by what they eventually describe. In particular, the relevant evaluation criteria are Paradigm, Formalism, Universality, Level, Probabilistic, Time progress and Time horizon (cf. Tbl. 2).

In the incubator example, the models emphasize the system's thermal behavior and control rules rather than its detailed structural attributes. The chosen paradigms of differential equations, state machines, and hybrid automata focus on capturing the incubator's physics and behavior by the dynamic temperature evolution affected by the heating element and user interventions. This approach is domain-specific, as it targets the thermal characteristics inherent to the incubator rather than adopting a single modeling paradigm or formalism for the entire incubator. Consequently, the incubator's models are positioned at a conceptual level that abstracts the control logic and thermal dynamics without detailing the complete physical implementation. They are deterministic and follow a fixed bang–bang control strategy, which accounts for time in a continuous fashion while being dynamic enough to accurately capture state changes over time.

It is important to recognize that modeling formalisms exist on a spectrum of formality from informal, to semi-formal, and formal [162–164]. This distinction is critical as it directly influences the reliability, reproducibility, and analytical power of a DT [137]. Informal models (e.g., sketches, natural language descriptions) are useful for initial brainstorming but lack the precision for automation or verification. Semi-formal models, such as standard UML [165] or SysML [166], possess a well-defined syntax but may have ambiguous semantics, making them excellent for design and communication but challenging for rigorous, automated analysis. Formal models (e.g., Petri Nets, Statecharts with formal semantics) are defined with mathematical precision in both syntax and semantics and enable unambiguous interpretation, formal verification, and proof of correctness. In the context of MBDTE, the chosen level of formality determines the degree of trust that can be placed in a DT's simulations, predictions, and control actions, thereby making it a crucial aspect of our evaluation.

Model Contents

Model characteristics focus on the contents of the models in what they describe, at what level for what purpose. In addition, this criterion group investigates the correctness of models. The criteria are Subject, Scope, Verification, Purpose, and Fidelity (cf. Tbl. 2).

In the incubator example, the model contents are designed to capture the system's behavioral dynamics. The models are part of both the DT's runtime as well as the MBDTE engineering process. Its scope is defined to encompass the cyber–physical interactions, viz. the exchange of heat governed by sensor readings and the actuator's bang–bang control strategy. To ensure reliability, the models are functionally verified, ensuring that they accurately depict the system's behavior and that the control rules correctly drive the system's response. The primary purpose of the models are twofold: the plant model facilitates simulation for real-time monitoring, and the coupled model supports decision-making for control actions such as reconfiguration when the lid is opened. Finally, a high degree of fidelity is achieved by using a physics-based plant model which is calibrated to match experimental data, and a controller model validated by refinement, which allows for a high-resolution and accurate exchange of information between the system and its DT.

Development

Finally, model development criteria investigate the techniques applied for maturing models. The criteria in this group, in particular, investigate the actual application of methods and techniques from MBE. This group's criteria comprise Construction, Assembly, Fusion, Management, and Modification (cf. Tbl. 2).

In the incubator example, model construction is done manually by domain experts who derive the differential equations following physical thermal dynamic principles. This thermal model is placed in a micro-service architecture to connect it with services representing configuration of the control logic, achieving non-hierarchical fusion. The resulting DT is not managed through a model repository or knowledge graph, though components can be assembled through a Digital-Twin-as-a-Service platforms [167, 168]. Finally, when adjustments are required, e.g., recalibrating parameter values or refining control thresholds based on observed performance, modifications can be made manually with semi-automated tool support. As well, the self-adaptation service can automatically adapt the system, when the lid is opened or when plant dynamics change substantially (e.g., an ice bucket is placed inside).

3.2.2 Data

Data criteria provide the tools for inquiring the incorporation of data during MBDTE, similarily to Combemale et al.'s earlier investigation of the different roles that data plays during software construction and operation in the context of MBE [128]. As argued in Sec. 2, data are the second key ingredient in DTs and thus merit dedicated investigation in the context of our evaluation criteria for MBDTE. In particular, these criteria investigate what kind of data is employed and how it is eventually integrated with the DT, i.e., uni- or bidirectional but also the actual coupling with models, DTs' most intriguing feature. In the following, the criteria groups *specification* and *integration* are described in more detail.

Specification

Data specification criteria query all aspects relevant to its nature, i.e., where it emerges from, how often it is available, what kind of data it is, and finally, whether it is uni- or multi-modal. These criteria can be distinguished into Data source, Data frequency, Data type, and Modality (cf. Tbl. 2).

In the incubator example, data comes from embedded sensors that continuously monitor parameters such as temperature and CO_2 (in newer versions of the incubator). The data frequency is high, with sensor readings delivered every second to capture fine-grained variations in the system's environment. The data types include real-valued measurements for thermal values, supplemented by binary values for whether the fan and heater are active. Additionally, the system employs a unimodal data approach focusing exclusively on environmental sensor input, although the framework is designed with the flexibility to incorporate multi-modal data as annotations from operator inputs or remote diagnostic tools.

Integration

Integration criteria are a key aspect in that they investigate a DTs' most intriguing feature of data integration. In particular, these criteria assess the actual type of twin (c.f., Fig. 3, p. 7) and how data is coupled with the models describing a DT's manifestation. Specifically, the criteria in this group are Direction and Coupling (cf. Tbl. 2).

In the tempeh incubator's DT, the coupling of data and models is designed to be bidirectional. Sensor data continuously flows into the model, enabling real-time monitoring and analysis of the system's environment. Simultaneously, the model outputs, which constitute control signal configuration, are transmitted to the actuators to adjust the incubator's conditions accordingly. The coupling between models and data is achieved by "feeding" data as input parameters into models. This connection is facilitated by the RabbitMQ message broker.

3.2.3 Implementation

A crucial aspect for the feasibility of MBDTE are necessary technological foundations for DTs to be realized. In that respect, implementation criteria investigate the current state-of-the-art regarding this realization. Consequently, this dimension has a dual purpose as it outlines the most common — or important, respectively — technologies, and on the other side, current technological limitations in that specific aspects of DTs currently lack realization. Such a lack becomes evident by correlating values from this dimension with other dimensions, in particular, *Characteristics* (c.f. Sec. 3.2.1) and *Evidence* (c.f. Sec. 3.2.4). The relevant evaluation criteria are: Framework, Technology and Language (see Table 2).

For the incubator example, sensor data is transmitted via RabbitMQ, . In addition, data is persisted in an InfluxDB for later analysis. Python is used for the sensor data processing and control logic, while a web-based dashboard provides real-time visualizations. Models are implemented in Modelica or Python. Docker is used to encapsulate and virtualize the various components.

3.2.4 Evidence

The efficacy and maturity of MBDTE is evaluated through the use of evidence criteria. These criteria encompass Maturity, Measures of evidence, the concrete Application/use case of the engineered DT (cf. Tbl. 2) and the relation to the life cycle Phase (cf. Fig. 4) of the real world entity. In speculative approaches, it is worth noting that the evidence criteria can be considered optional (and thus may not always contain a value after classification), as certain outcomes may not be readily attainable through industrial application and study.

The incubator example is a lab prototype and pedagogical tool [161], offering a testable solution for validation purposes. The system's maturity is evaluated using both qualitative assessment such as system responsiveness, and quantitative measures, including the alignment between sensor readings and model predictions. As a concrete application, the DT is utilized for real-time monitoring, control, and optimization of the incubator's environment. This approach is applied throughout the operational phase, where iterative adjustments are made to ensure optimal performance of the physical system.

3.3 Relation To Existing Studies

In light of the related studies discussed earlier (cf. Sec. 2.4), our criteria differ substantially. They are designed for evaluating MBDTE by establishing a holistic framework that explicitly connects the dimensions of models, data, implementation, and evidence. This stands in contrast to other frameworks reviewed in Sec. 2.4. Uhlenkamp et al. [155] focus on a maturity model with seven dimensions (context, data, computing, etc.), providing high-level stages of development but without in-depth analysis of model construction or data integration. Gil et al. [150] adopt an ISO-inspired framework for assessing open-source DT platforms, emphasizing system capabilities, architecture, and life cycle, but do not deeply investigate model-data integration or formal modeling languages. Autiosalo et al. [153] offer a feature-based framework that defines DT characteristics, focusing on functionalities rather than systematic engineering processes. Lehner et al. [84] map the application of MDE to DTs, concentrating on MDE techniques such as transformations and code generation, yet lacking our comprehensive evidence dimension. Hakiri et al. [157] emphasize DT challenges in networked environments but focus primarily on connectivity and security aspects rather than structured MBE. Zambrano et al. [156] propose a component-based approach for DTs with reusable building blocks but do not systematically assess how models are developed, verified, or integrated with data.

In contrast, our work uniquely incorporates evaluation criteria that explicitly tie together modeling paradigms, levels of abstraction, model-data integration, and development techniques, thereby offering a more rigorous and systematic lens for assessing DTE in the context of MBE. Tbl. 3 concisely correlates our work with related studies (cf. Sec. 2.4) by highlighting both overlapping elements and distinctive contributions, again exemplified by the incubator example from Sec. 3.2.

4 Selection and Evaluation of Publications

This section outlines the methodical process of choosing publications on MBDTE and categorizing them based on the previous section's criteria. We also outline our validation procedure that we applied to our initial taxonomy (cf. Usman et al. [158]). The potential risks to the accuracy and reliability of the selection and classification process are further examined.

4.1 Selection of Publications

In developing our search string, we adhered to the guidelines established by Kitchenham et al. [86, 87]. We thus performed iterative refinements of our search string to ensure it captured a broad spectrum of relevant literature within the domain of DTE and MBE. This process allowed us to effectively identify and include studies across multiple databases, ensuring a thorough and systematic exploration of the existing body of work. The final search string encompassed a broad range of terms: `((digital OR virtual) AND (model OR shadow OR twin OR thread OR replica)) AND model-(driven OR based) AND (engineering OR development)`. Tbl. 4 outlines the search parameters for the various databases used: For every database, we extracted the maximum allowed amount of papers.

Excluding papers was done in two iterations following the guidelines established by Kitchenham et al. [86, 87] for conducting systematic literature reviews to ensure a rigorous and systematic approach in selecting and categorizing publications on MBDTE. A first prescreening pass excluded papers that violated any of the following rules:

1. It is peer reviewed.
2. It comprises more than 4 pages.
3. It is written in English.
4. It is not a duplicate, i.e., the paper was found only once, because of the use of multiple libraries.
5. It is not a survey.
6. The title contains the term *model* according to our usage, and/or
7. The title contains the term *DT* (or a synonym of it, cf. Appx. A).

After this initial prescreening pass, in a second iteration, papers are excluded by *abstract* and *content*, i.e., if the abstract does not provide any indication to MBDTE, a paper is excluded. Finally, if reading is necessary as no exclusion could be done prior, a paper may finally be excluded by its content being off-topic for our study. Otherwise, if no exclusion criterion applies during both iterations, a paper is included in our evaluation. Fig. 6 outlines this process in detail.

4.2 Taxonomy Validation

Prior to conducting the final classification of papers we both evaluated our understanding of model and validated our initial taxonomy following Usman et al. [158]. To this end, we randomly selected 5% of the papers out of the 439 papers selected for full text reading (cf. Fig. 6) which amounts to 21 papers [1–21]. Eleven papers were immediately excluded due to falling outside the scope of this study, cf. MBDTE [11–21]. The remaining ten papers were included for classification [1–10] and employed for validating our initial taxonomy. As a consequence of this validation, we applied one change to the initially proposed taxonomy in that we decided not to further investigate the underlying implementation mechanics of model-data-integration (c.f. *Integration* dimension under *Data* in Tbl. 2) as this information was in fact not available from any of the publications. Also, the current criteria, viz. *Direction* and *Coupling* already catch the models-meet-data aspect with enough detail to allow for further analysis. Other than that, we kept our initial proposal resulting in the taxonomy as outlined in Tab. 2.

As to our understanding of what a model comprises in MBDTE, our definition (cf. Def. 1) aligns with what is communicated in literature [1–21] in that - if a paper is included in our study - the discussed models are readily captured by our definition (cf. engineering models).

Table 3: Correlation of our work with related work (cf. Sec. 2.4) with insights from the incubator example [160].

Dimension	Our Work	Related Work	Incubator Example
Model	• Detailed multi-aspect description (specification, contents, and development process) • Captures structural, temporal, stochastic and behavioral properties	• Broad classification (e.g., simulation, physical, geometric) • Standard MDE techniques without granular details	• **Our work:** Details that the incubator's twin is built manually using behavior differential equation models • **Related work:** Would likely label it as a generic simulation or physical model
Data	• Specifies data attributes (source, frequency, type, modality) • Emphasizes bidirectional, real-time synchronization	• Focuses on connectivity and standard protocols • Lacks details on data attributes and coupling mechanisms	• **Our work:** Explains that continuous sensor data (e.g., temperature) updates the DT and drives control signal configuration • **Related work:** Would only acknowledge sensor data in a generic sense
Implementation	• Details technological landscape (frameworks, programming languages, integration challenges) • Specifies choices like Python and Docker	• Offers a high-level overview of DT architectures (often ISO-inspired) • Omits detailed implementation choices	• **Our work:** Explains use of MQTT, RabbitMQ, Python, and a web dashboard for in-depth integration • **Related work:** Would likely only provide a general summary
Evidence	• Assesses system maturity (concept, lab prototype, production) • Examines qualitative and quantitative evidence across life cycle phases	• Emphasizes broad maturity models without a systematic evidence breakdown	• **Our work:** Identifies clear metrics (e.g., sensor-model alignment, system responsiveness) and classifies the incubator as a lab prototype • **Related work:** Would offer only a broad maturity classification

Table 4: Search parameters across different databases.

Database	Restrictions	Years	Sorting	Results Extracted
IEEE	Journals and Conferences	2000-2024	By relevance	First 2000 results
ACM	Research Articles	2000-2024	By relevance	First 1000 results
Elsevier	Research Articles	2000-2024	By relevance	First 991 results
Springer	Articles and Conference Papers	2000-2024	By relevance	First 2000 results
Google Scholar (via Harzing's Publish & Perish [169])	Excluded patents and citations	2000-2024	By relevance	First 989 results

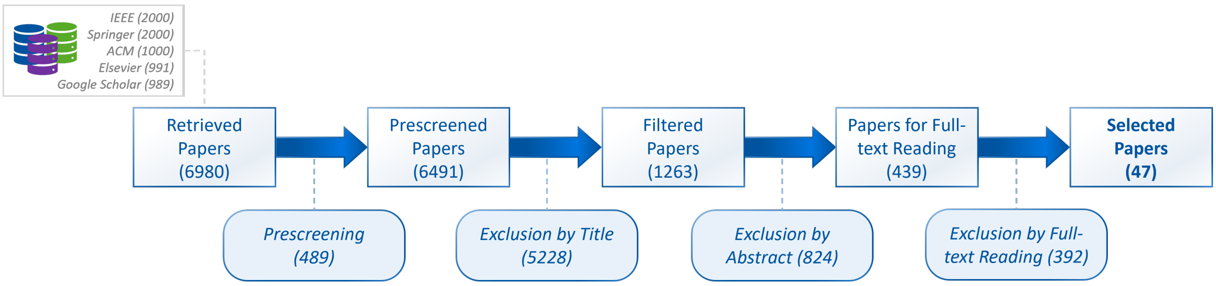


Fig. 6: Paper selection procedure.

4.3 Evaluation of Publications

The chosen 47 publications (cf. Fig. 6) on MBDTE were categorized based on the criteria presented in Sec. 3.1 by six researchers (the authors of this study) who also performed the selection process. To achieve this objective, the final set of publications (cf. Fig. 6) was randomly split into six groups of equal size for the purpose of extracting and classifying data. A spreadsheet was created specifically for this purpose. In addition to bibliographic details such as the title, year, and publisher, the form includes classification fields for each of the specified criteria (cf. Sec. 3.1). In regard to the exploratory aspect of our study, we deliberately avoided imposing any limitations on the input based on pre-compiled selection lists. This approach ensures that we investigate known research in an unbiased manner, without forcing it into a dedicated drawer. Conversely, our aim is to provide an unfiltered perspective on current studies. As a result, every publication underwent scrutiny from a minimum of three reviewers to ensure the accuracy of its categorization. In cases where a particular criterion was not clearly specified or could not be classified due to limited information in the publication, we reached a consensus to assign the value 'not available' (n.a.) to the field. Issues encountered throughout the classification process were noted in an additional comment section. All six researchers independently assessed the classification of the selected papers. Additional comments regarding the categorization of individual documents were also recorded in the comment field. Ultimately, at a collective session, all remarks were deliberated and settled among the entire cohort of six researchers participating in this investigation. During this session we also aligned the values in the spreadsheet's field regarding common typesetting (e.g., all lower or uppercase, using identical words instead of synonyms, ...) to ease subsequent analyses. Appx. B provides the list of papers that were included for our study.

Intended Limitations

Here, we only focus on MBDTE approaches with explicit models, i.e., machine-processable models. Our understanding of what a model has to represent in the context of MBDTE yields from a preliminary validation study, i.e., prior to evaluating any papers we randomly selected 5% of those papers selected for full text evaluation to infer a proper definition/usage of model from this set of papers. This validation study was conducted by three of the authors and subsequently evaluated by assessing its consistency in that a majority agreement has been reached as a consensus among the three validating authors.

In our evaluation and exclusion of MBDTE approaches we were liberal towards also including those that only yield a *digital shadow* [111]. The rationale is that our validation study and subsequent discussions revealed that it is often rather difficult to determine if an approach yields a *true* DT. This corresponds to earlier examinations of DT experience reports [152]. We justify our reasoning by considering Tao et al.'s service dimension in that *if* the virtual entity provides services for controlling or updating the real world entity – irrespective of being invoked manually of automatically – we agree on treating the engineering approach as an MBDTE approach given that it fulfills the other criteria (cf. Sec. 3).

5 Results

This section presents the classification results of the selected publications. The complete classification of all 47 papers using the proposed taxonomy is available for download [170]. In describing our results, we will list up 3-4 example papers for each category and sub-category even where there are significantly more papers in a particular category. We have chosen the example papers at random, the complete list of papers for each category is available from the aforesaid supplementary material and in Appx. B. To synthesize our findings, we performed frequency counts for each category and conducted cross-tabulations to analyze relationships and co-occurrences among the different criteria.

For each of the selected papers, it was possible to classify the covered MBDTE approach uniquely according to the defined criteria from Sec. 3. However, in performing our classification we often perceived a lack of detail in reporting [89] which for some categories resulted in a lack of evidence to infer a concrete statement (as will be pointed out during our upcoming discussions).

As we were able to fully evaluate the majority of selected publications, this indicates the adequacy of the defined criteria for the classification of MBDTE, thus providing a framework for understanding, categorizing, assessing, and comparing MBDTE approaches. Besides a validation of the proposed criteria, the systematic and comprehensive classification enables an aggregated view and analysis of the state of the art of MBDTE.

Figs. 7-9 and 14-16 show the detailed frequency counts of manifestations of the various dimensions and evaluation criteria of our taxonomy among the papers selected. Observe that frequency counts need not always add up to exactly 47 (cf. the number of selected publications) as a paper might report multiple values for a specific category.

5.1 RQ1: What is the current state of MBE for DTE?

Figs. 7-9 details the frequency count for the evaluation criteria under the *Model* dimension. The following discussions of RQ1 and its subquestions refer to these results.

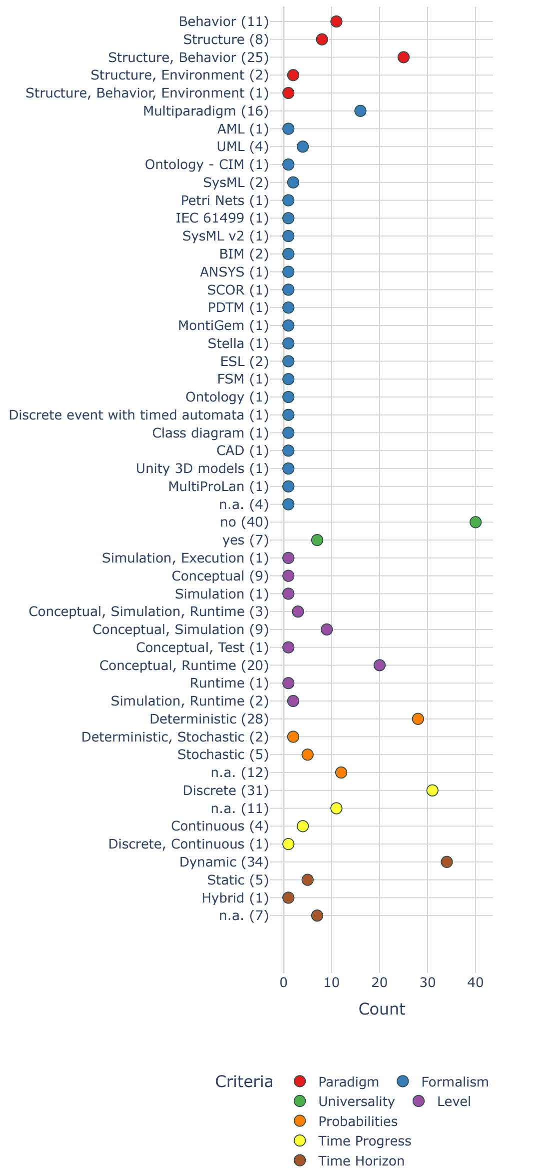


Fig. 7: Frequency counts of the criteria below the *Model Specification* dimension under the *Model* dimension as collected during evaluation of all 47 papers (cf. Sec. 4). Best viewed on a computer screen.

RQ1.1: What is the purpose of using MBE in DTE?

MBE serves a variety of purposes within DTE, most prominently *modeling* and *monitoring*. *Modeling* (16 occurrences; e.g., [5, 31, 44]), e.g., the engineering of the DT, is crucial for developing accurate DTs that can simulate real-world systems. It involves creating detailed representations that capture the complexity of physical entities and processes. *Monitoring* (21 occurrences; e.g., [28, 39, 57]), observing and synchronizing with the real-world entity's state, is another key purpose, where MBE facilitates tracking the performance and behavior of the DT versus the system in real time. This allows for ongoing assessments, adjustments, and optimizations, ensuring that the DT remains an accurate and reliable reflection of its real world entity.

Additionally, MBE is used for *prediction* (2 occurrences; e.g., [1, 35]) and *control* (11 occurrences; e.g., [4, 7, 26]), though these are less common use cases. Prediction and control represent runtime capabilities that are enabled by executable models within the DT. *Prediction* involves forecasting future states or behaviors of the physical twin based on a model, while *control* refers to the ability to influence the physical twin through the DT. Both employ runtime models (cf. models@run.time [118]) that enable prognostic analysis (predicting future states based on current observations) and adaptive control (generating commands to optimize behavior). These purposes illustrate the broad applicability of MBDTE in enhancing the functionality and value of DTs (cf. Fig. 8, *Purpose*). Our results at this point align with those of Dalibor et al. [149] in that monitoring is among the main applications for DTs. Specifically, we report 44.69% in our case vs. 29.95% for Dalibor et al. [149].

Our data reveals a clear hierarchy of adoption maturity. The dominance of monitoring as a use case demonstrates that the primary, industry-proven value of a DT today is in providing enhanced operational awareness, a powerful but eventually reactive capability. The significantly lower frequency of proactive applications like control and optimization suggests that the DT's role as an autonomous agent is still an emerging practice. This indicates that adoption is most likely to succeed by first targeting monitoring and diagnostics before progressing to more complex scenarios.

RQ1.2: How widespread—in terms of domains—is the use of MBE in DTE?

MBE is becoming a fundamental approach in DTE, particularly in complex domains like *CPS/-manufacturing* (26 occurrences; e.g., [26, 29, 33]). The fact that MBDTE is frequently applied in such specialized and complex environments suggests that it is well-suited to handle the challenges associated with engineering and operating DTs in these contexts. However, the distribution across domains is not entirely even, with some areas like *CPS/robotics* (3 occurrences; e.g., [44, 52, 53]) and *CPS/building* (4 occurrences, e.g., [27, 43, 48]) seeing less frequent use. This uneven distribution might reflect varying levels of maturity DTE or suitability of MBE across different industries (cf. Fig. 8, *Scope*).

RQ1.3: Does the use of MBE improve DTE and how?

MBE appears to contribute positively to DTE in several areas. The data shows a strong focus on *deterministic models* (30 occurrences; e.g., [6, 29, 33, 68]), which are crucial for efficient and understandable modeling of DT operations. These deterministic models allow engineers to create DTs that behave in a controlled and expected manner, reducing the likelihood of unforeseen issues. Additionally, the prevalence of *dynamic models* (34 occurrences; e.g., [3, 10, 30, 37]) highlights the importance of models that can adapt and evolve over time, mirroring the changes in the physical twin they represent. This dynamic capability is essential for maintaining the relevance and accuracy of DTs in real-time applications. The use of—as assessed during our evaluation—*high-fidelity models* (36 occurrences; e.g., [38–40]) underscores the relevance of MBE due to its capability of representing systems at different levels of fidelity (cf. Fig. 7, *Probabilities*/*Time Horizon*, and Fig. 8, *Fidelity*).

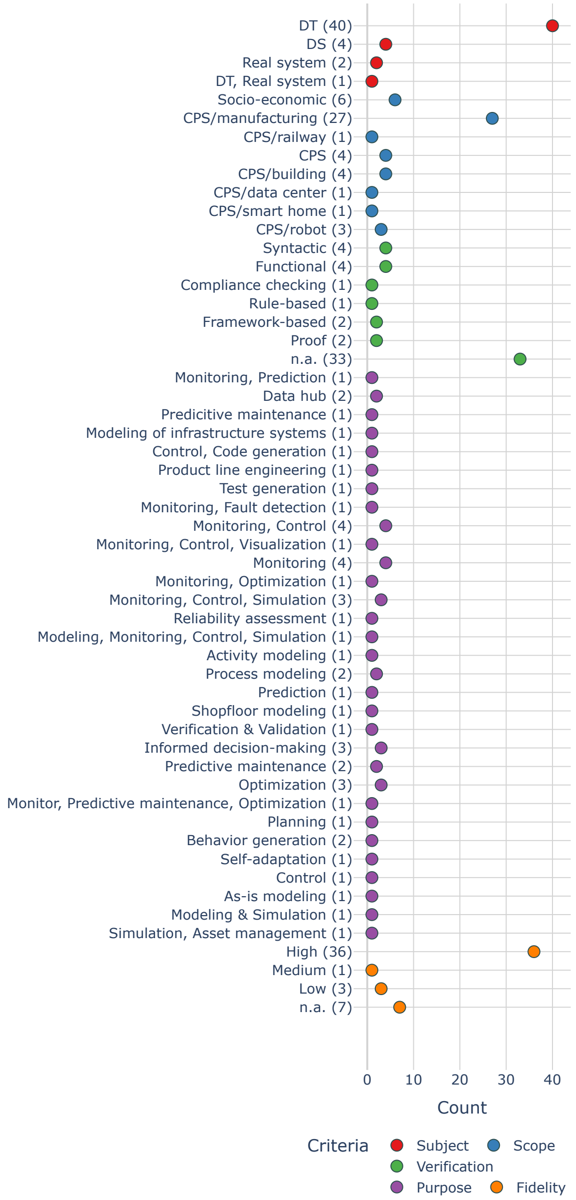


Fig. 8: Frequency counts of the criteria below the *Model Contents* dimension under the *Model* dimension as collected during evaluation of all 47 papers (cf. Sec. 4). Best viewed on a computer screen.

To decide whether a study reported a high-fidelity DT model, we introduced a three-level coding: High, Medium, Low, "not available". We applied this coding consistently across all 47 primary papers. A paper earned the *High* label only when it supplied explicit evidence of accuracy: either a quantitative claim (e.g., "$RMSE \leq 5\%$ against ground-truth sensor data") or a qualitative statement of physics- or component-level detail that was then validated or verified against measurements from the real system. Papers that described a potentially accurate model but offered no calibration or validation data were marked *Medium*, while purely conceptual or illustrative descriptions fell into *Low*. If no judgement was possible, we assigned "not available". In total, 36 studies (see above) meet our operational definition of high fidelity; however, the review found no empirical analysis linking fidelity levels to improved engineering outcomes, so the practical value of high fidelity remains unresolved.

The pattern of benefits demonstrates how MBE is currently leveraged in DTE: its primary contribution is to enable analysis and simulation, which positions a DT as a passive, decision-support tool. This indicates that while the benefit of using MBE to create models for the DT is well-established, its more advanced potential in providing models required for autonomous control remains largely unrealized.

RQ1.4: What is the level of maturity of MBE in DTE?

The maturity of MBE application in DTE (cf. Fig. 16) is predominantly at the prototype stage (32 occurrences; e.g., [7, 40, 44]), and to a lesser extent at the concept-study stage (14 occurrences; e.g., [8, 22, 29]). This distribution suggests that while MBE techniques are actively being researched and applied in DTE, their deployment in creating and maintaining DTs remains largely experimental. The reported prototypes and concept studies typically serve as proof-of-concepts that demonstrate the feasibility and potential benefits of applying MBE practices to DT development rather than representing mature, operationalized MBE workflows.

The limited number of mature implementations of MBE-driven DTs observed in our review suggests a gap between research on MBE methodologies and their full-scale industrial adoption in DT contexts; however, this may in part reflect our methodological focus on academic publications rather than industry reports. While our review indicates that MBE approaches for DTE often require further refinement and validation

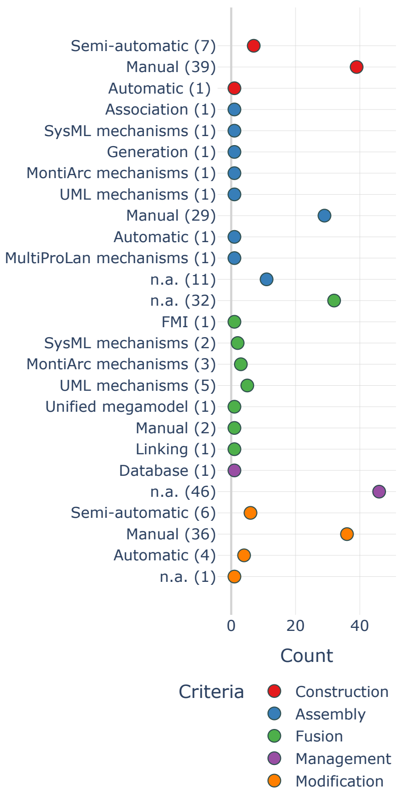


Fig. 9: Frequency counts of the criteria below the *Model Development* dimension under the *Model* dimension as collected during evaluation of all 47 papers (cf. Sec. 4). Best viewed on a computer screen.

before widespread deployment in productive environments, we acknowledge that mature DTs (as opposed to mature MBE processes for DTE) may exist in industry settings but employ MBE techniques that are not systematically reported in the academic literature. Examples in manufacturing [29, 31, 41] and rail systems [3] show DTs in applied settings that utilize model-based techniques, though the sophistication and scalability of the underlying MBE processes remain under-reported.

Critically, the concentration at the prototype stage indicates a significant gap between conceptual feasibility and production-grade deployment of model-based DTs. This suggests that current MBDTE methodologies may not yet adequately address the practical requirements for industrial-scale deployments, such as scalability, robustness, and maintainability, which remain a primary, unresolved challenge for the field (cf. Sec. 6.3).

RQ1.5: In which domains is MBE most frequently applied for DTE?

MBDTE has been applied across a variety of domains, with the most significant usage in *CPS/-manufacturing* (26 occurrences; e.g., [4, 23, 35]). This domain benefits greatly from the detailed and systematic approach that MBE offers, allowing for the creation of highly accurate and functional DTs that can simulate complex manufacturing processes. A reason for this strong adoption most likely can be found in existing standards that foster adoption of DTs in manufacturing [151, 171]. Other domains where MBDTE has been applied include *CPS/building* (4 occurrences; e.g., [27, 43, 48]), *CPS/robotics* (3 occurrences [43, 52, 53]), or *CPS/data center* (1 occurrence [30]). The application of MBDTE in these diverse fields demonstrates its versatility and adaptability. However, the lower occurrence in certain domains may indicate either a lesser need for MBE in those areas or a current lag in adoption and research in DTE (cf. Fig. 8, *Scope*). While our data reveals where MBDTE is most frequently applied, the reviewed literature provides limited empirical evidence regarding domain-specific benefits or comparative advantage of MBE across domains. Further research with rigorous evaluation and reporting metrics (cf. Gil et al. [89]) is needed to assess whether and how MBE effectiveness varies by application domain.

To answer RQ1, the current state of MBDTE is characterized by MBE's diverse use, especially in complex domains like CPS and manufacturing. The primary purposes of employing MBE in DT development include creating model artifacts (modeling) and enabling runtime state synchronization (monitoring). MBE further enables runtime prediction and control capabilities through executable models and models@run.time approaches. However, the maturity of MBDTE implementations remains largely at the prototype stage, indicating ongoing development and experimentation. MBE for DTE is being applied across various domains, though with varying degrees of adoption.

5.2 RQ2: How is MBE applied for DTE?

In answering RQ2 we again rely on the data presented in Figs. 7-9.

RQ2.1: What MBE concepts are used in DTE?

The data suggests that MBDTE relies heavily on concepts such as *high-fidelity modeling* (36 occurrences; e.g., [6, 35, 39, 43]) and *manual model construction* (39 occurrences; e.g., [2, 8, 37, 45]). High-fidelity models are essential for creating detailed and accurate DTs that can reliably replicate the behavior of physical twin. These models provide a rich source of data and insights, which can be used for simulation, monitoring, and control. The reliance on *manual construction* indicates that the development of these models is still a labor-intensive process, requiring significant expertise and careful attention to detail. This approach, while thorough, may limit scalability and increase the time required to develop and deploy DTs. There is also evidence of some *framework support* (15 occurrences; e.g., [5, 35, 45, 56]) in model management, which can help streamline processes and improve efficiency, though this is less common (cf. Fig. 8, *Fidelity*, and Fig. 9, *Construction*).

RQ2.2: What modeling formalism and paradigms are used in developing DTs?

Modeling formalism here refers to the formal language or notation used to define and describe models. It also defines syntax, semantics, and rules for constructing models (e.g., UML, SysML, BPMN, or DSLs). The data shows a significant use of *multi-paradigm* approaches (16 occurrences; e.g., [26, 45, 57, 58]) alongside *single-paradigm* models (27 occurrences; e.g., [9, 34, 38, 50], cf. Fig. 7, *Paradigm*). Among the 16 studies applying multi-paradigm modeling, the most common combination is structure models with behavior models, as illustrated by Fig. 10 (9 occurrences; e.g, [3, 26, 28]). Structure and environment models are combined only two times [27, 43], Grimmeisen et al. [9] even combine structure, environment, and behavior models.

The use of multiple modeling paradigms allows for more flexible and comprehensive modeling, accommodating the complexity of different systems and processes. This flexibility is crucial for creating DTs that can accurately reflect the diverse and interconnected nature of their physical twins and their environments. However, in some papers, single-paradigm modeling is applied to capture multiple model paradigms (e.g., UML in [38, 41], or SysML in [22, 37] for a combination of structure and behavior models). A more detailed discussion of the concrete technologies, tools, etc. used to implement models, not the formalisms or paradigms themselves, is deferred to the answer to RQ4.1 (cf. Sec. 5.4).

The diversity of formalisms highlights a critical trade-off in DT development. The prevalence of software-centric formalisms like UML/SysML is suited for describing system structure and control logic, but these often lack the capability to natively simulate complex (physical) dynamics. Conversely, physics-based formalisms offer high-fidelity simulations but demand specialized expertise and tools. This opposition implies that no *one-size-fits-all* modeling approach currently exists which forces engineering teams to make significant upfront decisions that might constrain the DT's capabilities and determine the required project skill set.

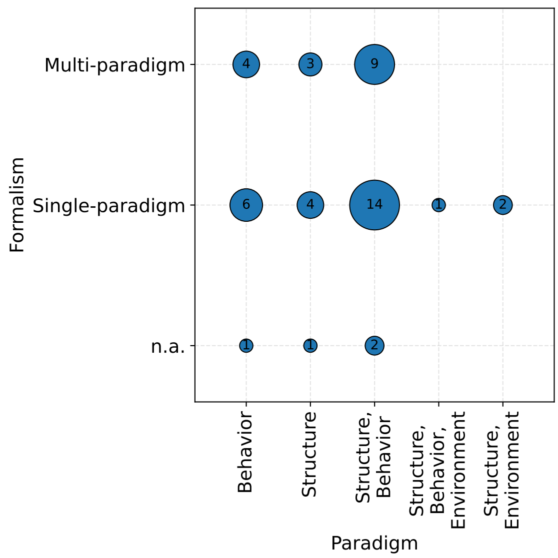


Fig. 10: Modeling formalism by paradigm in MBDTE. Best viewed on a computer screen.

RQ2.3: What is the level of abstraction of models?

The level of functional abstraction for models in MBDTE is predominantly *conceptual* (42 occurrences; e.g., [5, 9, 26, 41], cf. Fig. 7, *Level*). This high level of abstraction is necessary for dealing with complex systems that require a broad, system-wide view. Conceptual models (e.g., UML- or SysML-based [6, 8, 22, 37]) enable engineers to capture essential aspects of a system without implementation details, which can be particularly useful during early design stages (e.g., outlining requirements and architecture). However, the formalization overhead may not be justified for all DT systems, particularly simpler or rapidly-evolving prototypes. However, the challenge lies in ensuring that these high-level models remain sufficiently detailed to be useful for practical application down the engineering chain.

Fig. 11 demonstrates the levels of model abstraction in relation to the maturity of the study. It shows that conceptual models are mainly used in *concept studies* (6 occurrences; e.g., [22, 28, 32]), sometimes combined with runtime (6 occurrences; e.g., [35, 39, 57]) or simulation (only in [10]) models. In more mature implementations (*prototype*), conceptual models are more often combined with runtime (14 occurrences; e.g., [54–56]) or simulation (8 occurences; e.g., [52, 53, 58]) models, or both (3 occurrences; [4, 29, 34]).

The predominance of conceptual models creates an *abstraction gap* between design-time artifacts and operational executables. This necessitates manual translation, which severs the formal link between a system's design specification and its operational twin, eventually causing design models to to degenerate to static documentation rather than a living component.

RQ2.4: Are models characterized by static or dynamic properties, and do they exhibit stochastic or deterministic behavior?

Models in MBDTE are primarily *dynamic* (34 occurrences; e.g., [24, 29, 38, 47]) and *deterministic* (30 occurrences; e.g., [3, 10, 36, 55], cf. Fig. 7, *Probabilities / Time Horizon*). Dynamic models are crucial for capturing the time-dependent behavior of systems, allowing DTs to evolve alongside their physical twins. This is essential for applications that require real-time monitoring and control (e.g., [26, 30, 57]). Deterministic models, on the other hand, provide predictability and stability, ensuring that the outcomes of simulations and analyses are consistent and reliable. The combination of these properties indicates that MBDTE models are designed to be both responsive and dependable, and capable of adapting to changes while maintaining a high degree of accuracy.

In Fig. 12 it is shown that *dynamic* models are mostly used for modeling *behavior* (9 occurrences; e.g., [1, 7, 29]) or a combination of *structure, behavior* (21 occurrences; e.g., studies [3, 6, 22]). Finally, *environment* models are underrepresented and mostly unclassified, pointing to a potential area for further exploration in MBDTE.

RQ2.5: What do models in a DT represent: the physical twin, the computational environment/infrastructure, or both?

RQ2.5 investigates the representational scope of models within DT systems: whether they primarily capture the physical twin (its structure,

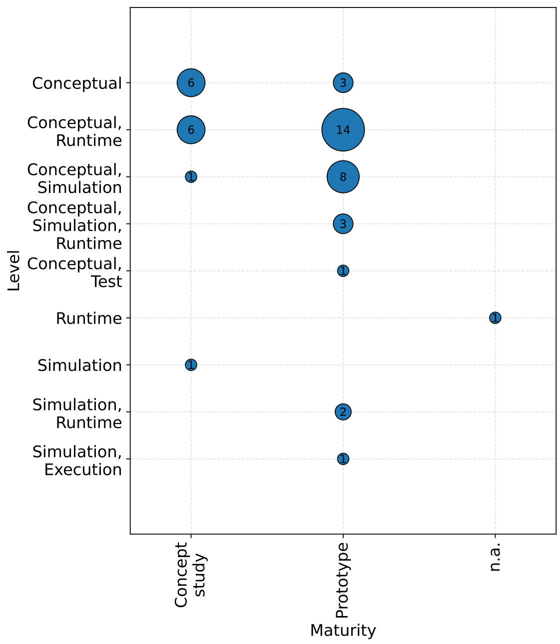


Fig. 11: Level of model abstraction by maturity of the study in MBDTE. Best viewed on a computer screen.

behavior, environment), the computational environment/infrastructure (execution platform, communication protocols, resource management), or both. This distinction is important because it reflects different modeling purposes, e.g., physics-based simulation versus system architecture and deployment, though in practice these concerns are often intertwined.

For instance, Chen et al. [40] outlines a four-layer architecture that demonstrates how models span both concerns: the model and simulation layer captures the dynamic behavior of the physical system (physics-based models), while the data layer and application layer implicitly rely on models of the computational infrastructure (data schemas, service interfaces, communication protocols). This structure epitomizes that most DTs employ models at multiple levels of abstraction to represent both the physical twin and the software system that implements the DT.

In most of the cases, models primarily represent the physical twin and its behavior: 41 studies (e.g., [40, 41, 45, 46]) employ models that capture the behavioral, interactive, and operational characteristics of the physical twin. Three studies [47, 48, 52] explicitly distinguish between models of the physical twin itself (component-level structural and process models) and models of the DT software system (data schemas, communication protocols, execution environment). This distribution shows that while most MBDTE research focuses on modeling the physical twin, the architectural and infrastructure concerns of the DT system itself receive less systematic modeling attention.

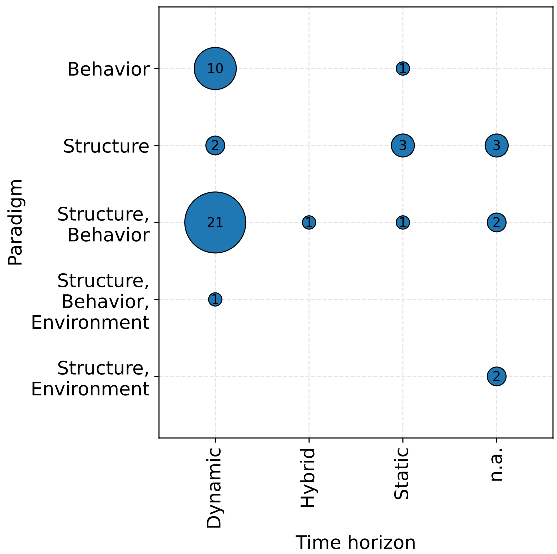


Fig. 12: Model paradigm by time horizon in MBDTE. Best viewed on a computer screen.

RQ2.6: What types of systems are modeled in which way, i.e., what types of models are used for what purpose?

Most studies rely on detailed, simulation-ready models that capture the underlying physics or system logic in enough depth to support decision-making and performance optimization (e.g., studies [2, 6, 7, 7, 30]). This diversity in modeling purposes reflects the complexity of the systems being studied and the need for tailored solutions to address specific challenges.

Beyond the dominant *CPS/manufacturing* cluster, smaller but noteworthy clusters are *buildings and infrastructure* (4 occurrences; e.g., [27, 43]), *robotics* (3; e.g., [44, 54]), *aerospace* (1; e.g., [37]) and specialized singletons such as *data-center* cooling or *smart home* assets (1 each; e.g., [30, 57]). Nearly all DTs are *dynamic* (34 occurrences) and operate in *discrete time* (32 occurrences; e.g., [28–30]), whereas continuous-time (5 occurrences; e.g., [6, 36, 53]) and stochastic (7 occurrences; e.g., [9, 39, 42]) representations remain the exception. This spread highlights both the field's manufacturing bias and the emergence of DTs in energy-intensive or safety-critical domains, as well as a prevailing preference for analytically grounded, event-driven models over continuous or probabilistic formulations.

RQ2.7: How is model correctness assured?

Model correctness in MBDTE is rather seldom assured (14 occurrences; e.g., [9, 45, 53, 54], cf. Fig. 8, *Verification*). This process involves careful checking and validation by experts to ensure that the models accurately represent the physical twins they are intended to describe. While manual verification is thorough, it is also time-consuming and may not be scalable for large or complex systems. The reported lack of evidence highlights a potential area for improvement, as more automated and scalable verification methods could enhance efficiency and reduce the risk of errors.

Recent advances in model checking [172], formal verification [173], and AI-assisted verification frameworks [174] have enabled automated approaches that can systematically explore large model state spaces and ensure compliance with formal specifications. These approaches reduce reliance on expert manual verification while increasing confidence in correctness and consistency. However, their adoption in MBDTE contexts remains limited [175], and further integration into existing toolchains is needed to realize their full potential. Recently, Munoz et al. [176] and Zech et al. [177] proposed two novel approaches for assessing the fidelity of simulations using similarity metrics. Munoz et al. capitalize on trace alignment whereas Zech et al. adapt the IoU[1] from computer vision to time series data. These approaches allow to quantify how well a simulation reflects the real system, ultimately providing insight into model validity and correctness in terms of how well real-world properties and phenomena are captured in model-based abstractions.

Our documented lack of formal verification is a critical risk indicator as it suggests that many DTs that are described in the literature may function only reliably under ideal conditions. Yet, their behavior is not guaranteed when faced with the unexpected or edge-case scenarios which are common in real-world operations or under what-if simulations. This gap in assurance impacts the trustworthiness of the DT and limits its use for high-stakes applications such as autonomous control or safety-critical decision-making.

RQ2.8: How are models developed and managed?

Model development in MBDTE is largely *manual* (39 occurrences; e.g., [2, 7, 43, 47], cf. Fig. 9, *Construction*), with some support from *frameworks* (15 occurrences; e.g., [4, 23, 53]). Manual development allows for a high degree of customization and attention to detail, but it also requires significant time and expertise. The use of frameworks can help streamline the development process and provide tools for managing models more efficiently. However, the data suggests that framework support is not yet widespread, indicating that many

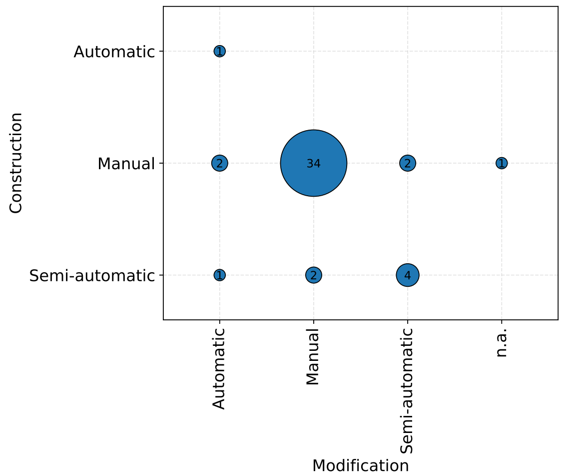


Fig. 13: Model construction by modification of models in MBDTE. Best viewed on a computer screen.

approaches still rely heavily on manual ad-hoc solutions and processes. For sure, another cause could also be that employed frameworks for now simply are not reported with enough detail.

The modification of models is usually also manual, provided that the construction is manual, as illustrated by Fig. 13 (34 occurrences; e.g., [42–45]). In merely two cases each, the modification is then done automatic [10, 54], or semi-automatic [40, 49]. With semi-automatic construction, the model modification is also semi-automatic in four ([1, 4, 5, 35]), manual in two ([53, 56]), and automatic in one case ([23]).

The finding that model development is overwhelmingly manual suggests an implementation bottleneck. Manual development risks creating tightly-coupled and brittle systems where the maintenance overhead of managing intermodel dependencies negates the intended benefits, pointing to a need for more formalized and automated model development techniques. In addition to this, the observed infrequency of model evolution points to a critical long-term risk: *model drift* [178]. Without systematic coevolution between virtual and physical instances, the DT's fidelity will degrade as the physical asset changes [114], which in turn threatens the long-term validity and reliability of its predictive and analytical capabilities. Without systematic processes, the DT risks becoming an outdated *digital*

[1] Intersection-over-Union

fossil rather than a living model, which leads to a gradual erosion of trust and a poor return on the initial investments.

RQ2.9: How are models combined?

When not manually performed, models in MBDTE are combined most often using modelling language mechanisms, such as those found in SysML (e.g., [22]), MontiArc (e.g., [5]), and UML (e.g., [46]). Megamodeling was also found to be used (e.g., [34]), which involves creating a model that encompasses and manages multiple sub-models, allowing for a cohesive and integrated approach to complex systems [179]. This method could be particularly useful in MBDTE, where systems often consist of multiple interconnected components that need to be modeled and managed together.

> To answer RQ2, MBDTE is applied using a combination of high-fidelity, deterministic, and dynamic models that are primarily developed manually. These models are deeply integrated into the DTs, serving purposes such as monitoring, control, and prediction. While the use of frameworks and advanced modeling techniques like megamodeling is emerging, much of the work is still done manually, particularly in the verification and management of models. This highlights the potential for further automation and integration of tools to enhance the efficiency and scalability of MBDTE.

5.3 RQ3: How are models and data coupled?

In answering RQ3 which targets the *models-meet-data* aspect, we rely on the data presented in Fig. 14. Specifically, Fig. 14 shows the frequency counts for the criteria under the *Data* dimension (cf. Tbl. 2) as collected during our analysis.

RQ3.1: What sources of data and what types of data are utilized?

The data utilized in DTE comes primarily from *sensors* (39 occurrences; e.g., [3, 5, 29, 33]), which provide real-time, continuous data streams that are essential for maintaining accurate DTs. This sensor data is typically *numeric* (32 occurrences; e.g., [2, 7, 10, 42]), making it well-suited for quantitative analysis and modeling. The reliance on sensor data highlights the importance of real-time information in keeping DTs synchronized with their physical twins. Additionally, the use of numeric data indicates a focus on measurable and precise inputs, which are critical for accurate simulations and control.

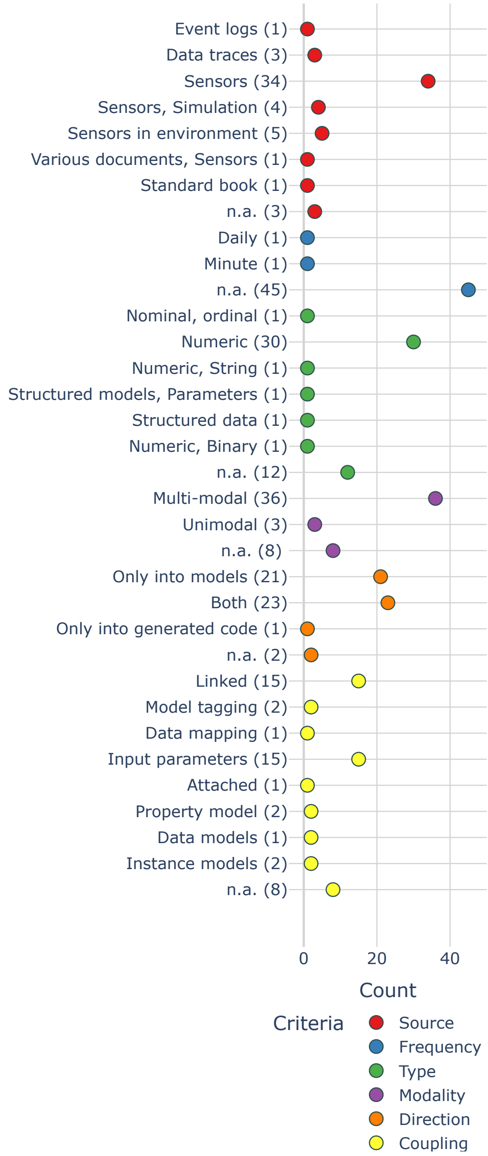


Fig. 14: Frequency counts for chosen values for the criteria of the *Data* dimension as collected during evaluation of all 47 papers (cf. Sec. 4). Best viewed on a computer screen.

The heavy reliance on structured and numeric sensor data highlights a critical limitation in the analytical scope of many current DTs. While they excel at processing telemetry and time-series data, they might struggle to integrate and reason with unstructured data sources such as maintenance logs, operator notes, or image data. This creates an *observability gap*, as contextual, often human-generated data required for complex root cause analysis usually lies outside the twin's perception, thus substantially limiting its diagnostic capabilities to what can be measured by sensors alone.

RQ3.2: Does data exhibit a uni- or multi-modal nature?

Data used in MBDTE often has a *multi-modal* nature (36 occurrences; e.g., [8, 9, 41, 57]), meaning that it includes multiple types of data from various sources. This multi-modal data can enhance the DT's richness and accuracy by providing a more comprehensive view of the physical twin. For example, integrating numeric sensor data with qualitative inputs or different data streams allows for more sophisticated modeling and better decision-making. The ability to integrate and analyze diverse data types is crucial for creating DTs that are both accurate and versatile.

RQ3.3: How often are models and data synchronized?

While the exact frequency of synchronization is not detailed in the provided data, the emphasis on *dynamic models* (34 occurrences; e.g., [6,10,57,58]) suggests that synchronization is likely an ongoing, real-time process. Dynamic models require continuous updates to reflect changes in the real world system, meaning that data and models must be regularly synchronized to maintain accuracy. This continuous synchronization is essential for applications where real-time monitoring and control are critical. Synchronization between models and data in a DT involves two key aspects: updating the data and updating the models. Data synchronization ensures that any significant change in the physical twin's state is promptly reflected in the DT. Strong emphasis (34 occurrences; e.g., [6, 10, 57, 58]) is observed on real-time synchronization, though the recommended frequency of updates largely depends on the nature of the physical system and its application domain. For example, Lie et al. [10] advocate for real-time synchronization in wind turbine structures to support precise control, while Barat et al. [6, 58] suggest that daily updates are adequate for tracking customer numbers in a communications service provider, since these figures do not fluctuate dramatically within a single day.

Model synchronization, on the other hand, is tied directly to changes or adaptations within the physical system itself. Essentially, whenever the physical twin undergoes modifications, the DT's model must also be updated to maintain reliability and accuracy. However, current literature does not offer concrete examples of adaptive model of DTs that evolve in tandem with their physical counterparts.

RQ3.4: How are models and data synchronized?

Our data shows that model and data synchronization generally is achieved, yet with only about 50% supporting synchronization in both directions. As to the concrete techniques used for achieving this synchronization, *linking models and data* (15 occurrences; e.g., [1, 4, 27, 31]) as well as *using data as input parameters for models* (15 occurrences; e.g., [9, 26, 29, 30]) are the prevalent approaches. The former is often achieved by hybrid modeling (e.g., [1, 4, 27, 31]), model calibration (e.g., [1, 31]), and data assimilation (e.g., [4, 31]). For the latter, e.g., model-to-model transformation [9], or rule-based model transformation [30] are applied. This suggests that while there are structured approaches to maintaining alignment between virtual and physical twins, full resolution of the models-meet-data challenge has not yet been achieved. The prevalent reliance on manual synchronization activities (e.g., *linking and tagging*; e.g., [4, 5, 7, 24]) may limit scalability and increase the risk of discrepancies between the DT and the physical twin, particularly in large or complex environments.

The reliance on linking and tagging for model-data synchronization suggests that many DTs operate with significant latency and function closer to *Digital Shadows* (cf. Fig. 3) than real-time counterparts. This gap between envisioned autonomous data flow and observed methods poses a fundamental limitation for applications requiring high-frequency, closed-loop control.

RQ3.5: Do models only consume data or also emit data, e.g., control values?

Following on our answer to RQ3.4, models consequently not only consume data but also emit data, e.g., in the form of *control values* (11 occurrences; e.g., [9, 25, 26, 30]) or *predictions* (2 occurrences [1, 35]) for informed decision-making. This bidirectional data flow is crucial for the interactive nature of DTs, where the model can influence the physical twin through feedback loops. For example, control values generated by a model can be used to adjust the operation of the physical twin, ensuring optimal performance or responding to changes in the environment. This interactive capability renders models a powerful tool for real-time system management.

The predominant lack of bidirectional data flow means that most DTs function as advanced, *read-only* monitoring systems rather than proactive control agents. This limits a DT's role to that of a sophisticated advisor that can flag problems, but prevents them from acting on those insights to (autonomously) optimize or correct the physical asset's behavior.

To answer RQ3, models and data in MBDTE are closely integrated, with a strong reliance on real-time sensor data and multi-modal data sources. Synchronization is an ongoing process, particularly for dynamic models that require continuous updates. Models in DTE are not only consumers of data but also emit data, enabling them to interact with and control real world systems. However, the reliance on manual processes for synchronization and model management suggests opportunities for further automation and scalability improvements.

5.4 RQ4: How is MBE implemented in DTE, specifically what frameworks, technologies, development languages, and modeling tools are used?

For answering RQ4, in addition to the data from Figs. 7-9 we additionally consulted the data shown in Fig. 15 which shows the frequency count of our

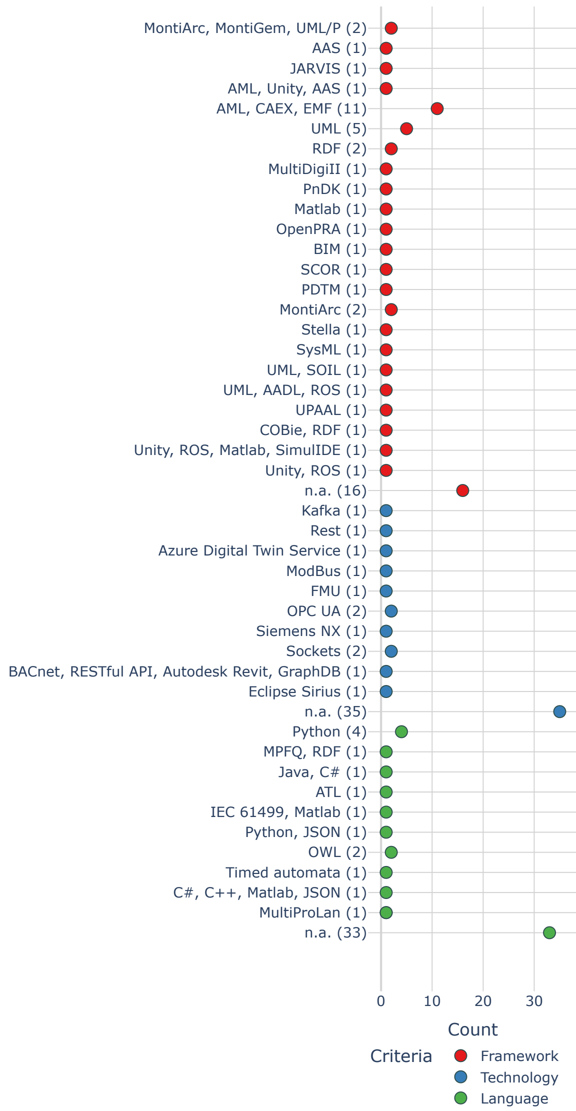


Fig. 15: Frequency counts for chosen values for the criteria of the *Implementation* dimension as collected during evaluation of all 47 papers (cf. Sec. 4). Best viewed on a computer screen.

categories under the *Implementation* dimension as collected during our analysis.

The implementation of MBDTE involves the use of established frameworks like UML and tools such as MATLAB. UML, a standardized modeling language, offers a general-purpose visual modeling framework for system design, which is why it appears in nine occurrences across various studies (e.g., studies [6, 8, 44, 57]). Its application in MBDTE reflects a preference for well-structured and widely understood frameworks capable of supporting complex system modeling. On the other hand, MATLAB, mentioned in two occurrences [30, 53], is a high-level language and interactive environment predominantly used for numerical computation, visualization, and programming. Its use suggests that numerical analysis and simulation are crucial components of MBDTE implementations, providing the computational power necessary for developing and testing high-fidelity models.

However, the use of such frameworks is not without challenges. For instance, MATLAB presents integration challenges with other tools, necessitating the development of robust communication interfaces to manage data flow effectively [53]. Additionally, MATLAB's limitations in real-time data processing and lack of native multi-threading hinder efficient communication [53], highlighting the complexity of employing MATLAB for MBDTE in large-scale, dynamic systems [30].

Similarly, using UML also presents several challenges related to complexity, integration, and scalability. UML's inherent complexity can make managing large systems difficult, especially when modeling intricate interactions and dependencies within DTs [8]. Integrating UML models with other tools and platforms requires additional effort to ensure compatibility and effective communication between different system components [44]. Moreover, maintaining scalability while using UML to model dynamic environments and synchronize DTs with their physical counterparts can pose significant challenges, particularly when dealing with real-time data and interactions [57]. These challenges underscore the need for careful planning and execution when employing UML in DT projects [6].

Regarding development languages, our data is somewhat sparse, with approximately 70% of publications not reporting on the used languages. A similar trend is observed in the reporting of modeling frameworks and technologies, with 16 and 32 works not specifying these details, respectively. Nonetheless, from the available data, there is a slight preference for versatile and widely supported development languages like Python (found in five occurrences, e.g., studies [1, 29, 55]) and *C#* (found in two occurrences, viz. studies [25, 53]). Python is known for its ease of use, extensive libraries, and cross-domain applicability, making it a popular choice for developing DTs and related models. *C#* is another powerful language often used in environments requiring integration with legacy technologies or specific software ecosystems.

The use of specialized tools like SimulIDE [53] and Eclipse Sirius [56] reflects the need for environments that support the design, simulation, and management of DTs and their models. These tools provide essential interfaces and functionalities to develop, test, and deploy complex models in an integrated manner.

We also note that well-known DT implementation frameworks such as Eclipse Ditto and Azure DTs do not appear in our survey. This may be due to the model-based focus of our survey. Gil et al. provide a survey of DTs utilizing these frameworks [150].

The technological fragmentation of DT implementations is not merely a tooling issue; it creates a more profound architectural deficiency. The lack of consensus on frameworks and languages directly prevents the emergence of a standardized *models and data backbone* [114, 180] for structuring and exchanging DT data. Without a common foundation for semantic interoperability, DTs are often implemented as isolated data silos, a structure which fundamentally impedes higher-level applications like the composition of system-of-systems DTs [181]. From a project management perspective, this technological fragmentation translates directly into higher project risk.

To answer RQ4, MBDTE implementations rely on a mix of traditional frameworks like UML and tools and languages such as MATLAB and Python. The choice of languages and tools reflects the need for versatility, computational efficiency, and integration capabilities. This combination allows engineers to develop and manage complex models that are essential for DT applications, though there is room for more standardized and scalable tooling to support the growing complexity of MBDTE.

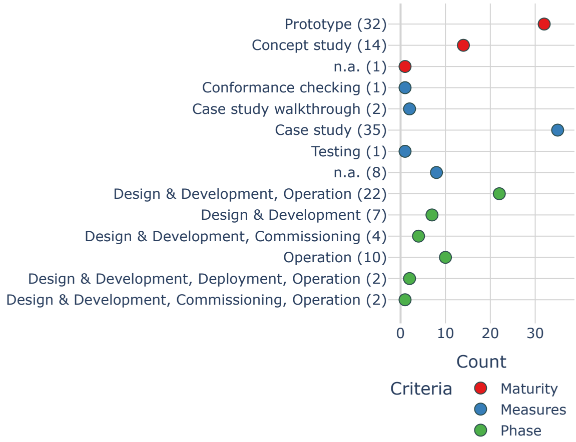


Fig. 16: Frequency counts for chosen values for the criteria of the *Evidence* dimension and the phases in which a DT is employed as collected during evaluation of all 47 papers (cf. Sec. 4). Best viewed on a computer screen.

5.5 RQ5: In which life cycle phases of a DT is research located?

For answering our remaining RQs in addition to the data from Figs. 7–9, 14, and 15, we also rely on the data outlined in Fig. 16, which shows the frequency count of the categories under the *Evidence* dimension in our analysis. Observe that as to its diverse spread, the *Application* criteria is not reported here but available from the full classification [170].

The research and application of MBDTE is primarily concentrated in the *Design & Development* phase (37 occurrences; e.g., [1, 9, 22, 49]), with significant activity also observed in the *Operation* phase (36 occurrences; e.g., [1, 9, 29, 51]). The focus on *Design & Development* highlights the importance of creating accurate and functional DTs from the outset, with careful attention to model fidelity, integration, and scalability. This phase is critical for setting the foundation of the DT, as decisions made here will impact the DT's effectiveness and longevity.

The high activity in the operation phase underscores the ongoing nature of DT management, where continuous monitoring, control, and updating are required to maintain accuracy and relevance. This phase is where the dynamic nature of DTs comes into play, as they must adapt to changes in the real world entity and its environment. The presence of research in these phases indicates that MBDTE is seen as an ongoing process, essential not just for initial development but also for the long-term success and utility of DTs.

Our survey reveals a gap in the intermediate life cycle phases of commissioning and deployment. This oversight implies a lack of model-driven methodologies for critical activities like calibration and validation, and represents a significant practical barrier to the seamless commissioning and operationalization of DTs.

5.6 RQ6: What new MBE research challenges are created by its application in a DT context?

The emergence of MBDTE has introduced several new MBE-related research challenges to be addressed as part of future research agendas (cf. Sec. 6) for successful advancement, including:

- **Scalability of Model Development and Management**: As DTs become more complex, the manual processes currently used for model development, verification, and synchronization may become untenable. Our analysis revealed that 39 studies (e.g., studies [8, 24–26]) reported the use of manual model construction (cf. Fig. 9) —a key hindrance for scalable model development and management. The development automated or semi-automated tools could help manage the complexity and scale of DTs more effectively.
- **Model-Data Synchronization**: Ensuring real-time synchronization between models and physical twins is critical but challenging. Despite a variety of approaches to model-data synchronization (36 studies, e.g., [9, 10, 28, 31], cf. Fig. 14), real-time synchronization is a key limitation in efficient model-data synchronization. None of our analyzed studies address the real-time aspect. Our results suggests that integration of diverse data types is attempted, robust methods for continuous real-time synchronization are however needed to maintain model accuracy and fidelity.
- **Integration of Multi-modal Data**: The ability to integrate and analyze diverse data types from various sources is essential for creating accurate and versatile DTs. Our results confirm

this in that multi-modal data integration is well employed (36 studies, e.g. [31, 33, 35, 38]), yet there is a lack in reported consistency in data quality and relevance indicating that additional work is necessary to ensure that integrated data maintains high quality and supports accurate DT functionality.

- **Enhancing Model Fidelity**: While high-fidelity models are already in use, there is still room for improvement, particularly in terms of balancing model complexity with simulation efficiency. High-fidelity models are reported in 36 studies (e.g., studies [34, 37, 38, 42], cf. Fig. 8), indicating their importance. However, inconsistencies in how fidelity is achieved and reported suggest that there is room for improvement. Future research should focus on optimizing model complexity and simulation efficiency to enhance fidelity further.
- **Verification and Validation**: As models become more complex, ensuring their correctness and reliability through manual verification processes may not be sufficient. Only 15 studies (e.g., studies [42, 49, 53, 54], cf. Fig. 8) mention that they do verification. This highlights a significant gap in ensuring model correctness and reliability. Developing automated verification and validation methods could provide more robust assurances of model accuracy.
- **Framework Support and Standardization**: The current use of diverse tools and frameworks suggests a need for more standardized approaches to MBDTE [110]. Our data shows a wide range of frameworks being used, with 16 different frameworks mentioned (e.g., studies [4, 23, 28, 30], cf. Fig. 15), demonstrating a lack of standardization. There is thus a clear need for unified frameworks that can provide comprehensive upport across the DT life cycle, enhancing interoperability and adoption across domains.

Sec. 6 more thoroughly discusses future research challenges with a potential guide for action, aligned with our answers to RQ6.

5.7 RQ7: What aspects of MBDTE merit deeper investigation and more thorough reporting in literature?

Our analysis reveals several areas where current literature on MBDTE is lacking detail, which suggests opportunities for deeper investigation and more thorough reporting [89] as part of future research agendas (cf. Sec. 6). To assess the current approaches to MBDTE, the literature needs to provide more comprehensive case studies, cross-domain comparisons, and longitudinal studies. Additionally, there is a need for standardized evaluation metrics [182], deeper exploration of emerging technology integrations, and a focus on human factors [58, 183], security, scalability, and interoperability. Addressing these gaps will contribute to a more thorough understanding of the effectiveness and challenges of MBDTE, ultimately leading to more robust and scalable solutions. Specifically, following our analysis, we identify the following suggestions for advancing research in MBDTE:

- **Detailed Case Studies and Practical Applications**: While many papers discuss theoretical frameworks and methodologies, there is a noticeable gap in detailed case studies that explore practical implementations of MBDTE. While 35 out of the 47 studies reviewed, provide a "case study" (e.g., studies [7, 25, 26, 29], (cf. Fig. 15, *Measures*) for evaluation, our analysis revealed that these are rather "case" discussions but lack the rigor and depth classical applied for case studies to sufficiently study a contemporary phenomenon [184]. This indicates a critical gap in practical application insights. Actual case studies would provide valuable insights into the challenges, successes, and lessons learned from real-world applications.
- **Cross-Domain Comparisons**: The research largely focuses on specific domains, but there is a lack of comparative studies that analyze how MBDTE is applied across different fields. Our data shows a strong focus on specific domains like CPS/manufacturing (26 studies, e.g. [9, 39, 45, 50], (cf. Fig. 8, *Scope*), with significantly fewer studies exploring cross-domain applications. Understanding how MBDTE techniques differ or overlap in various sectors could

lead to the development of more versatile and widely applicable methods.

- **Longitudinal Studies on MBDTE Maturity**: The current literature provides snapshots of the state of MBDTE, but there is a scarcity of longitudinal studies that track the evolution of MBDTE. The maturity of MBDTE is predominantly at the prototype stage (31 studies, e.g. [52, 54–56], (cf. Fig. 15, *Maturity*), with limited longitudinal data on the evolution of these approaches. Such studies however would offer insights into the maturity of MBDTE approaches, highlighting trends, advancements, and areas that require further development.
- **Comprehensive Evaluation Metrics**: There is a need for more comprehensive and standardized metrics to evaluate the effectiveness of MBDTE. While some papers provide qualitative assessments, a rigorous approach to quantifying the benefits and limitations of MBE would help in objectively assessing its impact on DTE. Specifically, 14 papers provide concept studies (e.g., studies [8, 10, 28, 33]) and 31 (e.g., studies [52, 54–56]) offer prototype evaluations. This shows there is a clear need for standardized, comprehensive metrics to rigorously quantify the impact and limitations of MBDTE like the recently published *Overall Digital Twin Entanglement* (ODTE) metric [182].
- **Integration of Emerging Technologies**: Although there is some discussion about integrating technologies like AI, IoT, and machine learning with MBE and especially DT architectures [5, 185], the exploration of these integrations in the engineering process is often superficial. Our analysis reveals that so far no efforts have been undertaken in integrate IoT-based communication aspects and AI support in the engineering processes in MBDTE. Deeper investigation into how these technologies can be effectively combined with MBE and MBDTE to enhance DT capabilities is necessary.
- **Human Factors and Usability**: The literature tends to focus heavily on technical aspects, often overlooking the human factors involved in implementing and using MBDTE. Our analysis and data reveal a lack of attention to the skills and training required for effective MBDTE implementation, which is crucial for broader adoption. Understanding the usability challenges faced by engineers and operators, as well as the training and skills required, is crucial for successful adoption.
- **Security and Ethical Implications**: There is limited discussion on the security and ethical implications of using MBDTE. Our analysis and data demonstrate a substantially limited discussion on security and ethical issues, suggesting areas that need more exploration. As DTs become more integrated into critical infrastructure, it is essential to explore how MBDTE can address concerns related to data security, privacy, and ethical decision-making.
- **Scalability and Performance in Large-Scale Systems**: While some papers touch on the scalability of MBE approaches, there is a lack of in-depth studies on how MBDTE performs in large-scale, complex systems. Our data reveals a lack of in-depth, longitudinal studies on large-scale MBDTE performance. With the increasing complexity of DTs, research into the scalability of models and tools, as well as data management, is essential for expanding MBDTE applicability outside lab conditions.

Sec. 6 will seize these answers in providing a detailed discussion on future research challenges with a potential guide for action.

6 Discussion

This section synthesizes and interprets the results of our work. We begin by providing a consolidated summary of our findings in response to the research questions (cf. Sec. 6.1). We then contextualize these contributions by positioning them against related work in the field (cf. Sec. 6.2). Building on this analysis, we identify and elaborate on open research challenges (cf. Sec. 6.3) and conclude by addressing the threats to the validity of our study (cf. Sec. 6.4).

6.1 Summary of Findings

Our exploration of MBDTE reveals a rapidly evolving field with significant potential but also notable challenges. The current state of MBDTE (RQ1) shows that while MBE is being increasingly adopted across various industries, its application is not yet widespread. MBE is used primarily to enhance system design, optimize performance, and support decision-making processes within digital twins. However, the maturity of MBDTE practices varies greatly, with more advanced implementations found in industries like aerospace and automotive, while other sectors are still in the early stages of adoption. The benefits of MBDTE, such as improved reliability, cost efficiency, and predictive maintenance, are recognized, but the data suggests that these advantages are not yet fully realized universally.

The purpose of using MBDTE (RQ1.1) is largely driven by the need for more accurate and dynamic system models that can adapt to real-time data and changing conditions. However, the extent of MBE's use in DTE (RQ1.2) is still limited, with many approaches experimenting with pilot projects rather than full-scale implementations. The data also indicates that while MBE does improve DTE in certain areas (RQ1.3), such as predictive maintenance and system optimization, these improvements are often contingent on the maturity of the MBE tools and methodologies used. The maturity of MBDTE (RQ1.4) remains a key challenge, with significant variability across different sectors and applications. MBDTE has been applied in a wide range of domains (RQ1.5), including manufacturing, healthcare, and infrastructure, but the depth and sophistication of its use differ greatly.

When examining how MBE is applied in DTE (RQ2), it becomes clear that a diverse range of MBE concepts, formalisms, and paradigms are employed. The models used in MBDTE vary significantly in their level of abstraction (RQ2.3), from high-level conceptual models to detailed, domain-specific models. These models may exhibit static or dynamic properties (RQ2.4) and are used to represent deterministic or stochastic behaviors. Importantly, models can either be an integral part of the DT or serve as supporting infrastructure (RQ2.5). The types of systems modeled and the purposes of these models are diverse (RQ2.6), reflecting the broad applicability of MBDTE in different industries. Ensuring model correctness (RQ2.7) – if done – is mainly achieved through automated tooling, although this remains an area needing further research. Models are typically manually developed and managed (RQ2.8), and there is a perceivable increased interest in the ways models can be combined – ideally automatically – to create more comprehensive and integrated DTs (RQ2.9).

The integration of models and data in MBDTE (RQ3) is another critical area. The types and sources of data utilized (RQ3.1) are varied, ranging from sensor data to historical performance data, and can be uni- or multi-modal in nature (RQ3.2). Synchronization between models and data (RQ3.3) is crucial for maintaining the accuracy and relevance of the DT, with methods ranging from real-time synchronization to periodic updates (RQ3.4). In some cases, models do not just consume data but also generate outputs, such as control signals (RQ3.5), which are then fed back into the physical twin.

The implementation of MBDTE (RQ4) involves a wide range of frameworks, technologies, and tools. There is no single dominant framework. Instead, a variety of development languages and modeling tools are used (RQ4.2), reflecting the diversity of applications and requirements in DTE. The life cycle phases in which MBDTE research is located (RQ5) tend to focus on early-stage development, such as design and testing, but there is growing tendency in extending MBDTE practices to the operational and maintenance phases of DTs.

New challenges for MBE in the context of DTE (RQ6) are emerging as the complexity of DTs increases. These include the need for better tools for model verification and validation, improved real-time synchronization techniques, and the integration of emerging technologies such as AI and machine learning. There are also gaps in the current literature (RQ7), particularly in terms of practical case studies, cross-domain comparisons, and the development of standardized evaluation metrics.

While the field of MBDTE is advancing, there remain significant challenges and areas for further research. The potential benefits of MBDTE are clear, but realizing these benefits on a broader scale will require continued innovation, particularly in the areas of tooling, automation, standardization [110], and the integration of emerging technologies. Addressing these challenges will be crucial for the continued evolution and success of digital twin engineering.

Finally, we remark that the relatively small number of publications employing MBE for DT development (47 papers) compared to the broader DT literature suggests that MBE is neither universally adopted nor universally necessary. Many successful DTs are developed without formal modeling approaches. Our survey focuses specifically on cases where MBE *is* applied, and characterizes the benefits and challenges in those contexts rather than advocating for its universal adoption.

6.2 Positioning To Related Work

In what follows, we correlate our observations with earlier discussed related work (cf. 2.4) to position our contributions within the broader field of MBDTE.

6.2.1 Convergence of MBE and DTE Practices

A key challenge discussed by Dalibor et al. [149] and Gil et al. [150] is the lack of integration between MBE and DTE frameworks. Both studies underline the need for a stronger convergence between the modeling methodologies of MBE and the dynamic, real-time requirements of DTE.

Our research builds directly on this foundation by also investigating on *Model Fusion* and *Model Fidelity*, two aspects frequently overlooked in prior works. While Gil et al. [150] emphasize the need for a standardized framework for DTs, our findings from Sec. 5 show that MBDTE approaches must incorporate more precise and dynamic models to fully capture the evolving state of physical twins, echoing Dalibor et al. [149].

6.2.2 Addressing Reporting Gaps

A recurring challenge, highlighted by Gil et al. [89] and further substantiated by Oakes et al. [152], is the pervasive lack of detailed reporting on pivotal elements such as modeling structures, model fidelity, and synchronization methods. Our study echoes these concerns, revealing that a significant portion of the literature falls short in providing comprehensive documentation of these critical aspects. This gap in reporting not only obscures the science but also severely undermines the reproducibility and comparability of research outcomes, creating significant barriers to progress and innovation in the field of MBDTE.

Insufficient reporting on key aspects like modeling structures, model fidelity, or synchronization makes it exceedingly difficult to replicate studies or benchmark them against one another. Without clear, comprehensive insights, researchers are left navigating a fog, unable to effectively build on existing work or adapt methodologies to new contexts. This lack of clarity ultimately decelerates the pace of advancement in MBDTE, as novel ideas remain siloed and unvalidated across studies.

Our approach provides a systematic framework for evaluating MBDTE approaches, aiming to dispel this fog. By offering a detailed taxonomy that encompasses model, data, implementation, and evidence aspects, we not only enhance the reporting framework suggested by Gil et al. [89] but also set a higher standard for transparency and rigor in the field. This focus on detailed, standardized reporting is not merely a call for academic thoroughness; it is a critical step toward ensuring that research in DTs can be reliably replicated and compared, thereby unlocking their full potential across diverse domains. This emphasis on rigorous reporting is essential for fostering an environment of collaboration and innovation, where advancements can be effectively shared and built upon.

6.2.3 Model Fidelity and Integration Challenges

Another critical aspect is the challenge of combining various models and maintaining fidelity over time. Dalibor et al. [149] and Autiosalo et al. [153] examined the limitations in cross-domain modeling and suggested that many DT solutions rely heavily on physics-based models without adequately incorporating real-time analytics for adaptive behavior. Our findings in Sec. 5 support this claim, as we observed that most papers (34 out of 47) employed static models, failing to leverage more dynamic, real-time data coupling methods.

Moreover, Sec. 6 shows that despite the availability of powerful, general-purpose modeling formalisms (e.g., UML), the practical fusion of models with real-time data – essential for the dynamic nature of digital twins – remains underdeveloped. Only a handful of publications explored this critical aspect, confirming the gaps pointed out by Dalibor et al. [149] regarding the fusion and modification of models in real-time environments.

6.2.4 Maturity and Industrial Application

Sec. 5 further reveals that the maturity level of most MBDTE implementations remains at the prototype stage. This observation is consistent with the assessments made by Uhlenkamp et al. [155] in their maturity model for DTs, which also noted a lack of fully operational DT systems, especially in cross-domain applications like manufacturing and urban planning. Our findings highlight that while MBDTE holds promise for life cycle management and predictive maintenance, real-world applications still lag behind in integrating models with real-time feedback loops.

In summary, our study not only reaffirms the gaps identified in existing literature, such as inconsistent reporting and limited model-data integration but also advances the discussion by proposing a structured framework for improving reporting standards in MBDTE. By comparing our results with the works discussed in Sec. 2.4, it becomes clear that while there has been progress in understanding the theoretical underpinnings of MBDTE, there is a pressing need for more rigorous reporting and industrial-scale implementations to fully realize the potential of DTs in various domains.

6.3 Open Research Challenges

We conclude this section by providing a thorough overview of future research challenges lying ahead as well as potential actions to resolve those to successfully advance MBDTE. Our following discussions further reflect our perspectives on our answers to RQ6 in Sec. 5.6.

Challenge A: Enhancing Model Integration and Interoperability. One of the key challenges in MBDTE is the integration and interoperability of models across different tools, platforms, and domains. As DTs grow in complexity, the need to combine models from various engineering disciplines (e.g., mechanical, electrical, software) into a cohesive, interoperable system becomes critical [110, 112, 185–187]. Current MBDTE approaches often struggle with this due to the lack of standardized modeling languages and tools that can seamlessly communicate with each other. This reflects the findings of RQ6 showing a wide tool heterogeneity (i.e., 16 different frameworks) and the lack of standardization (cf. Sec. 5.6), and RQ7 pointing to missing cross-domain comparisons that would expose interoperability problems (cf. Sec. 5.7).

Actions: Promoting the development and adoption of standardized modeling languages and protocols, such as SysML [188, 189] or Functional Mockup Interface (FMI) [190], or approaches for meta-data exchange such as the AAS [191, 192], facilitate model exchange and interoperability across different tools and domains. Developing frameworks that can automatically integrate heterogeneous models allows to ensure that these models work together seamlessly within a DT. These frameworks should focus on resolving compatibility issues and maintaining consistency across the integrated models (i.e., see [193–195] for consistency checking and model linking approaches). Finally, collaboration between different engineering disciplines should be fostered [110, 196] to develop unified modeling approaches that can address the needs of various domains, to ensure that DTs are truly holistic representations of complex systems.

Challenge B: Improving Real-Time Model Updating and Synchronization. DTs rely on real-time data to accurately reflect the state of their physical twins. However, maintaining real-time synchronization between models and physical twins is challenging, particularly when dealing with complex, dynamic environments. While platform-level solutions especially for data synchronization exist [197–201], MBDTE research has not systematically addressed mechanisms to update models in real-time, leading to discrepancies between the DT and the physical twin. This reflects RQ6, where real-time synchronization was identified as missing across all reviewed studies (cf. Sec. 5.6). Moreover, real-time updating ties to challenge D on scaling and challenge F on integration of emerging technologies.
Actions: Research should focus on developing algorithms and methods for real-time model updating that can incorporate live data streams into the DT without compromising model accuracy or performance. This could involve advanced filtering techniques, real-time data assimilation, and adaptive modeling approaches. This in turn demands for methods for improving the flexibility of models so that they can adapt to changes in the physical twin dynamically. This could include developing models that are capable of self-updating based on incoming data or using machine learning techniques to predict and adjust

models for future system changes. Collectively this calls for robust synchronization protocols that ensure data from various sources is consistently and accurately reflected in the models.

Challenge C: Ensuring Model Accuracy and Validation. Ensuring the accuracy and validity of models in complex, evolving DT systems [187, 202–204] is a significant challenge in MBDTE. As models become more intricate and data-driven, the potential for errors increases, making it difficult to maintain high levels of confidence in the DT's predictive capabilities. Current validation and verification techniques may not be sufficient to handle the complexity and dynamic nature of DTs' models [84, 187, 205]. As emphasized in answering RQ6, only a small set of studies reported verification and validation activities (cf. Sec. 5.6) or evaluate model correctness (see RQ2.7), and answering RQ7 expresses the need for comprehensive evaluation metrics (cf. Sec. 5.7).
Actions: Automated tools that can rigorously verify model correctness, particularly for large and complex systems are paramount. These tools should be capable of detecting inconsistencies, errors, and potential failures in models before they are deployed. This readily forms a foundation for establishing continuous validation processes that regularly test and validate models against real-world data, ensuring that the DT remains accurate over time. This could involve real-time simulations or digital-physical feedback loops to constantly assess model correctness and performance. This requires to expand model-based testing for comprehensive testing of models under various design and operating scenarios, including edge cases and unexpected conditions. This, in turn, will help identify potential issues that could impact the reliability of the DT.

Challenge D: Scaling MBE for Large-Scale DTs. As DTs are applied to increasingly large and complex systems, such as smart cities or large industrial plants [84, 149, 183, 206], scaling both MBE approaches and DT architectures becomes a critical challenge. Traditional MBE tools and methods may not be capable of handling the vast amount of data and the intricate interactions within these large-scale DTs. This aligns with RQ6, which identified scalability of model development and management as a challenge (cf. Sec. 5.6), and RQ7, which reports that scalability and performance in large-scale systems are insufficiently addressed in literature (cf. Sec. 5.7).
Actions: Research should focus on creating scalable modeling techniques that can handle the complexity and size of large-scale DTs. This could involve modular modeling approaches, where the system is broken down into smaller, manageable components that can be modeled and updated independently but have to be integrated again [185]. In tandem, high-performance computing resources need to be leveraged to manage the computational demands of large-scale DTs during their runtime. Research into parallel processing techniques and distributed computing could enable the real-time analysis and updating of extensive DT models. Further, integrating cloud computing platforms in model-based approaches to support the scalability of MBDTE demands a more thorough investigation. Cloud-based solutions could provide the necessary infrastructure to store, process, and analyze large amounts of data from complex systems, ensuring that DTs and their DevOps process remain responsive and accurate.

Challenge E: Enhancing Usability and Accessibility of MBE. The complexity of MBE tools can be a barrier to their widespread adoption in DTE, particularly among non-experts. Many existing tools require specialized knowledge and significant expertise to use effectively, limiting their accessibility to a broader audience [187, 207]. This can hinder the integration of MBE into organizations that could benefit from DTs. This is supported by the findings in RQ7, showing that human factors and usability are underexplored aspects in MBDTE research (cf. Sec. 5.7).
Actions: Develop more intuitive tooling for MBDTE that can be easily used by engineers and other stakeholders without extensive training. This could involve creating graphical/visual modeling environments [208–211], generation of visual representations [212–214], low-code development environments [4, 215–217], or co-creational and collaborative modeling approaches [218, 219], or drag-and-drop interfaces [220] that simplify the building and managing of models. Alongside improving tool support, comprehensive training

programs and educational resources [221] to help engineers and managers understand and apply MBDTE are required. These resources should cover the fundamentals of MBE and DTE, as well as advanced techniques for model development and integration, cf. MBDTE. Last but not least, the development of collaborative platforms that allow multi-disciplinary teamwork merits substantially more attention than it receives now [114, 115]. These platforms should support real-time collaboration, version control, and easy sharing of models and data across organizations.

Challenge F: Integrating Emerging Technologies with MBDTE. Integrating emerging technologies such as AI, IoT, and machine learning into MBDTE offers significant opportunities for innovation, but it also presents challenges in terms of compatibility and complexity. Current MBE frameworks may not be fully equipped to leverage these technologies effectively, limiting their potential impact on DTE. This matches the gaps identified in RQ7, where integration of emerging technologies such as IoT, machine learning and AI was noted as largely missing in current research on engineering DTs (cf. Sec. 5.7).
Actions: Hybrid modeling approaches that combine traditional MBE techniques with AI and machine learning require increased investigation [222, 223]. These approaches could enable DTs to learn from data, adapt to changes, and optimize their performance over time. In addition, methods to better integrate IoT devices into MBDTE workflows [224], allowing real-time data from sensors and other devices to directly influence model behavior are beneficial. This could involve creating new protocols for data exchange and processing within DT environments. Eventually, development of AI-driven MBE and MBDTE tools, respectively, that can automate parts of the model development process, such as generating models from data or optimizing model parameters. These tools could significantly reduce the time and expertise required to build and maintain complex DTs.

Challenge G: Standardized Reporting for MBDTE Research. Throughout the course of our study, we frequently encountered situations where we were unable to effectively assess key criteria. These criteria include, but are not limited to, the specific frameworks utilized, the programming languages employed, the underlying technologies adopted, as well as critical aspects such as model fusion techniques and model fidelity. This lack of detailed and standardized reporting creates significant barriers in the field of MBDTE research. It not only limits our understanding of existing methodologies and their implementations but also severely hinders the reproducibility of results. Reproducibility is a cornerstone of scientific progress, and without clear documentation of the approaches taken, future researchers face considerable challenges when attempting to build on or verify the work of others. This absence of transparency makes it difficult to evaluate the robustness, scalability, or applicability of a given approach to other contexts. The answers to RQ7 highlighted this challenge by stressing the lack of rigorous cases studies, standardized evaluation metrics, and longitudinal studies (cf. Sec. 5.7). Moreover current research lacks to systematically report on areas where MBE can support DTE. One can find approaches where MBE enhances *Real-Time Monitoring and Control* [7, 26, 30, 31], improves *Predictive Maintenance* [35, 39, 41, 55], supports *System Optimization* [29, 40, 43], fosters *Design Flexibility and Innovation* [38, 46, 54], and improves *Cost Efficiency* [28, 41, 48], however, evaluations on the effects of MBE application on industrial KPIs are rarely included.
Actions: We contend that implementing more rigorous and consistent reporting practices would substantially advance research within the MBDTE domain. By fostering greater clarity and transparency, such standards would enable researchers to more accurately compare methods, reproduce results, and accelerate innovation. In turn, this would contribute to the development of more robust and reliable digital twin technologies, facilitating their broader adoption across industries. This could be achieved, for example, by consolidating Gil et al.'s work with ours. The 21 characteristics which were reported by Gil et al. [89] and provided tool support by Fiter *et al.* [213] provide a top-down, high-level view on the conceptual structure of a DT. On the contrary, our taxonomy with its substantially more detailed view provides the complementary bottom-up view by investigating implementation aspects of a DTs application.

Addressing the MBE-specific challenges for DTE is crucial for advancing the field of MBDTE. By focusing on improving model integration, real-time updating, validation, scalability, usability, and the integration of emerging technologies, researchers and practitioners can ensure that MBDTE meets the demands of increasingly complex and dynamic systems. These efforts will enhance the effectiveness of DTs and broaden their applicability across diverse industries and use cases. While acknowledging the challenges and gaps in implementing MBDTE, it is equally important to highlight the significant potential and benefits MBDTE. MBDTE revolutionizes the design and management of complex systems by providing a unified and integrated engineering workflow that enhances understanding, collaboration, and communication among stakeholders [225]. This not only facilitates development and automation but also optimizes operational efficiency and streamlines workflows. Furthermore, MBDTE supports innovation through virtual prototyping and formal validation and testing, enabling faster and more cost-effective development cycles. By overcoming these challenges, the transformative power of MBDTE can be fully realized, driving advancements across industries and creating new opportunities for innovation, growth, and success.

6.4 Threats to Validity

Naturally, our study - as is the case for any taxonomy proposal and literature review - is exposed to various threats regarding its validity and sound conduction. In the following we address these threats and outline how we mitigated them.

6.4.1 Internal Validity

Internal validity refers to the level of rigor and correctness in which a review combines and analyzes the research contained, in order to reach findings that are valid and dependable. To ensure internal validity, it is necessary to employ a thorough search strategy, maintain consistency in data extraction, and conduct a rigorous quality assessment of the studies that are included. Internal validity threats in systematic literature reviews encompass selection bias, data extraction mistakes, and publication bias, all of which have the potential to distort the review's conclusions.

Selection Bias

The process of choosing research papers to include in a review may exhibit bias. Insufficient comprehensiveness or transparency in the search strategy, inclusion, and exclusion criteria may result in the omission of relevant research or the inclusion of irrelevant ones. To effectively handle this issue, we utilized several digital libraries, implemented a systematic search-and-exclusion process, employed multiple reviewers, and conducted a control search using Google Scholar.

Classification Bias

Papers can be misclassified as a result of subjective interpretation of the taxonomy. This occurrence can arise when the taxonomy lacks clear definition or when the researchers possess divergent views of the classification criteria. To prevent this vulnerability, we adhere to established methods while building our taxonomy (see Sec. 2.3.1 for details) and verify the classification findings by cross-checking.

Publication Bias

Studies that yield good results are more prone to publication, resulting in an over-representation of certain findings in a systematic review. To effectively alleviate selection bias, it is crucial to ignore the “positiveness” or “negativeness” of a publication. By relying on multiple reviewers for classification and cross-checking, selection bias is substantially attenuated.

6.4.2 External Validity

External validity in a review refers to the extent to which the findings of the review can be applied to wider contexts beyond the specific studies that were assessed. To achieve high external validity, it is necessary to include a comprehensive and inclusive sample of research that encompasses a wide range of individuals, settings, and conditions. The presence of this diversity guarantees that the inferences derived from the review can be universally applicable and are not constrained by particular situations or populations. In addition, precisely defining the extent and constraints of the review assists in establishing the parameters within which the findings can be applied, hence improving the practical significance and usefulness of the review.

Generalizability

If the scope of the review is too limited or if the taxonomy utilized is too specific to a particular domain, the conclusions may not be relevant or applicable to other contexts or domains. To obtain a high level of generalizability, we implemented a thorough and all-encompassing search approach. We did not limit the application areas and used meticulously established search and exclusion criteria. Finally, we ensured transparency in reporting our findings. Furthermore, as a means of verifying our work, we performed a subgroup analysis using randomly chosen articles from different application domains.

Sample Representativeness

If the sample of papers used in the review does not accurately represent the total relevant literature, the findings may not be applicable to a broader context. In order to address this potential danger, we implemented a thorough and all-encompassing search strategy, established precise and criteria for inclusion and exclusion, and performed a subgroup analysis (cf. Sec. 4.2). Furthermore, our endeavors to reduce publication bias effectively decrease the risk of erroneous representation of the sample.

6.4.3 Construct Validity

Construct validity in a review refers to the degree to which the review accurately captures and measures the theoretical constructs it intends to investigate (cf. RQs, cf. Sec. 1.3). Achieving high construct validity involves clearly defining the key concepts and ensuring that the included studies adequately address these constructs through appropriate operational definitions and measurements. It also requires a careful selection of studies that consistently and accurately measure the constructs of interest. Ensuring construct validity strengthens the credibility and relevance of the review's conclusions by ensuring it genuinely reflects the theoretical concepts being studied.

Taxonomy Validity

The taxonomy itself may be inherently faulty. If the classification of articles fails to accurately represent the dimensions or categories that are important in the subject, it will result in a false and misleading classification. To address this issue, we employ established practices in developing a taxonomy (cf. Usman et al. [158]).

Misclassification

There is a possibility of ambiguity in the taxonomy, which can result in inconsistent classification. Divergent interpretations of taxonomy among researchers may result in disparate classifications for a given collection of publications. We resolve these issues by implementing the *four-eyes* approach, which requires that a particular activity, such as a decision, must receive approval from at least two researchers. By enforcing each publication to be reviewed by at least two researchers and the final cross-checking of classification results by all six involved researchers, misclassification is reduced substantially.

6.4.4 Conclusion Validity

Conclusion validity in a review pertains to the extent to which the conclusions derived from the synthesis of studies are believable and well-founded. To ensure conclusion validity, it is necessary to perform statistical analysis, thoroughly evaluate the quality of the study, and provide clear and open reporting of the review process. Additionally, it is necessary to acknowledge and rectify any potential biases and limitations present in both the included papers and the review itself. High conclusion validity assures that the findings and interpretations are strong, dependable, and accurately represent the evidence collected from the literature.

Data Extraction Errors

Errors in data extraction from the research can result in inaccurate conclusions. Ensuring consistency in the methods used to extract data and verifying the accuracy of the extracted data are of utmost importance. To mitigate this concern, we implemented a protocol where each of the 47 publications included for categorization underwent a rigorous evaluation process by a minimum of two researchers.

Interpretation Bias

Researchers' assumptions or previous conceptions can introduce bias when interpreting classified data. By adhering to the *four-eyes* principle and engaging in several rounds of group discussion, we significantly reduce the presence of interpretation bias to a minimal extent.

Conclusion Validity

The quality robustness of our conclusions is inherently linked to the quality and depth of the papers included in our survey. While we have diligently applied specific criteria to filter publications, our influence over the inherent quality and detail of these papers remains limited. A significant concern in this context is the potential lack of detail in one or more of the categories we propose, which could result in our survey not fully capturing the current state of the art in the field.

6.4.5 Additional Threats

Apart from these "usual" threats to validity, our study is subject to additional threats, as briefly discussed in the following.

Reviewer Expertise

The expertise and familiarity of the reviewers with the subject matter can impact the accuracy of the classification. Inexperienced reviewers might misclassify papers. We argue that this threat is of low relevance because the authors have combined multi-year experience in working in and the promotion of the MBE, DTE, MBDTE fields through various initiatives [226–230].

Inter-rater Reliability

Low inter-rater reliability suggests that different reviewers may categorize the same papers in varying ways. This difficulty can be reduced by employing many reviewers and utilizing techniques to address differences, including the aforementioned "four-eyes" idea and conducting multiple rounds of discussion to settle concerns over inter-rater reliability.

Taxonomy Evolution

In the event that the taxonomy undergoes changes during the review process, it can result in discrepancies in the classification of papers. We effectively addressed this concern by implementing Usman et al.'s [158] recommendation and conducting a validation study of our taxonomy before classifying any of the ultimately chosen papers (cf. Sec. 3.1).

7 Conclusion

This article has presented a comprehensive and systematic review of the current landscape of MBDTE, leveraging a taxonomy-driven analysis of 47 research publications. Our findings demonstrate that while DTs have gained significant traction across multiple industries—including manufacturing, healthcare, and urban planning—the adoption of MBE principles for the development and management of DTs remains in an emergent stage. This is evidenced by a predominance of prototype-level research and a limited number of fully operational, real-world implementations.

A central insight from our survey is the persistent gap in the practical integration of models and data — a defining feature of DTs. Most reviewed approaches fall short of realizing continuous, real-time synchronization and high-fidelity coupling between models and data. Although MBDTE shows promise for advancing complexity management, real-time monitoring, and predictive maintenance, substantial challenges remain, particularly regarding model-based monitoring, synchronization mechanisms, and the rigorous validation of high-fidelity models.

Crucially, our work foregrounds the importance of systematic and transparent reporting in MBDTE research. The lack of detailed reporting on key engineering choices such as update frequency, data-model coupling strategies, or tool-specific implementations still limits reproducibility, comparability, and, ultimately, the broader synthesis of best practices. This shortcoming constrains the ability of practitioners and researchers alike to draw empirically grounded conclusions that inform evidence-based engineering decisions.

Our proposed taxonomy provides a structured framework for classifying and evaluating methodological and application-specific aspects in MBDTE research. While we observed exemplar cases where reporting meets a high bar for rigor and detail, many publications offer only fragmentary accounts. This variance underscores a critical need: for MBDTE to mature as both a scientific and engineering discipline, future research should prioritize comprehensive, consistent, and context-aware documentation, particularly when aspects such as update frequency or validation metrics are central to the research aim.

Looking ahead, overcoming these challenges, especially in terms of reporting and integration, will be essential to extend MBDTE impact and foster its broader industrial adoption. By adopting more rigorous reporting standards and addressing current gaps in implementation evidence and evaluative criteria, the MBDTE community can advance towards robust, scalable, and trustworthy digital twin technologies across varied application domains. Our findings, taxonomy, and recommendations together provide a foundation for systematic progress and a benchmark for future empirical research in this fast-evolving field. While our survey demonstrates the potential of MBE for managing complexity in DT development, we emphasize that MBE is not universally required. The decision to adopt model-based approaches should be grounded in project-specific factors including system complexity, safety requirements, team structure, and available expertise.

Acknowledgments. The authors thank Claudio Gomes for his valuable input on the incubator example.

Appendix A DT Synonyms

- twin
- digital clone
- digital counterpart
- digital replica
- virtual replica
- virtual twin
- cyber twin
- digital copy
- mirror image
- simulated counterpart
- twin model
- virtual doppelganger
- virtual double
- virtual representation
- cyber replica
- digital doppelganger
- digital double
- holographic duplicate
- mathematical counterpart
- mirror twin
- simulated twin

Appendix B Included Papers

[1] Pan, Y., Zhang, L.: A BIM-data mining integrated digital twin framework for advanced project management. Automation in Construction **124**, 103564 (2021)

[2] Becker, F., Bibow, P., Dalibor, M., Gannouni, A., Hahn, V., Hopmann, C., Jarke, M., Koren, I., Kröger, M., Lipp, J., Maibaum, J., Michael, J., Rumpe, B., Sapel, P., Schäfer, N., Schmitz, G.J., Schuh, G., Wortmann, A.: A Conceptual Model for Digital Shadows in Industry and Its Application, pp. 271–281. Springer, Cham (2021)

[3] Zhou, S., Dumss, S., Nowak, R., Riegler, R., Kugu, O., Krammer, M., Grafinger, M.: A Conceptual Model-based Digital Twin Platform for Holistic Large-scale Railway Infrastructure Systems. Procedia CIRP **109** (2022)

[4] Dalibor, M., Heithoff, M., Michael, J., Netz, L., Pfeiffer, J., Rumpe, B., Varga, S., Wortmann, A.: Generating customized low-code development platforms for digital twins. Journal of Computer Languages **70**, 101117 (2022)

[5] Kirchhof, J.C., Michael, J., Rumpe, B., Varga, S., Wortmann, A.: 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 (MODELS). ACM, New York, NY, USA (2020)

[6] Barat, S., Kulkarni, V., Kumar, P., Bhattacharya, K., Natarajan, S., Viswanathan, S.: Towards effective design and adaptation of csp using modelling and simulation based digital twin approach. In: Summer Simulation Conference (2020)

[7] Zheng, X., Psarommatis, F., Petrali, P., Turrin, C., Lu, J., Kiritsis, D.: A Quality-Oriented Digital Twin Modelling Method for Manufacturing Processes Based on A Multi-Agent Architecture. Procedia Manufacturing

51, 309–315 (2020)

[8] Brecher, C., Dalibor, M., Rumpe, B., Schilling, K., Wortmann, A.: An Ecosystem for Digital Shadows in Manufacturing. Procedia CIRP **104**, 833–838 (2021)

[9] Grimmeisen, P., Wortmann, A., Morozov, A.: Case study on automated and continuous reliability assessment of software-defined manufacturing based on digital twins. In: 25th Int. Conf. on Model Driven Engineering Languages and Systems: Comp. Proc. (MODELS). ACM, New York, NY, USA (2022)

[10] Liu, Y., Zhang, J.-M., Min, Y.-T., Yu, Y., Lin, C., Hu, Z.-Z.: A digital twin-based framework for simulation and monitoring analysis of floating wind turbine structures. Ocean Engineering **283**, 115009 (2023)

[11] Bevilacqua, M.G., Russo, M., Giordano, A., Spallone, R.: 3D reconstruction, digital twinning, and virtual reality: Architectural heritage applications. In: IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), pp. 92–96 (2022). IEEE

[12] Shabani, A., Skamantzari, M., Tapinaki, S., Georgopoulos, A., Plevris, V., Kioumarsi, M.: 3D simulation models for developing digital twins of heritage structures: challenges and strategies. Procedia Structural Integrity **37**, 314–320 (2022)

[13] Xia, J., Huang, R., Chen, Z., He, G., Li, W.: A novel digital twin-driven approach based on physical-virtual data fusion for gearbox fault diagnosis. Reliability Engineering & System Safety **240**, 109542 (2023)

[14] Bazaz, S.M., Lohtander, M., Varis, J.: 5-dimensional definition for a manufacturing digital twin. Procedia Manufacturing **38**, 1705–1712 (2019)

[15] Zhenzhen, C., Wenzhong, T.: A behavior-based dynamic trust model of virtual organization. In: 3rd Int. Symposium on Information Science and Engineering, pp. 391–395 (2010). IEEE

[16] Gillespie, S., MacCalman, A.: A case study in developing an integrated data and model management system for the development of a complex engineered system. In: IEEE Technology and Engineering Management Conference (TEMSCON), pp. 1–6 (2018). IEEE

[17] Papageorgiou, N., Verginadis, Y., Apostolou, D., Mentzas, G.: A collaboration pattern model for virtual organisations. In: Leveraging Knowledge for Innovation in Collaborative Networks: 10th IFIP WG 5.5 Working Conference on Virtual Enterprises (PRO-VE). Proc. 10, pp. 61–68 (2009). Springer

[18] Catarci, T., Firmani, D., Leotta, F., Mandreoli, F., Mecella, M., Sapio, F.: A conceptual architecture and model for smart manufacturing relying on service-based digital twins. In: IEEE Int. Conf. on Web Services (ICWS), pp. 229–236 (2019). IEEE

[19] Yuan, J., Tian, Z., Ma, J., Man, K.L., Li, B.: A Digital Twin Approach for Modeling Electrical Characteristics of Bifacial Solar Panels. In: Int. Conf. on Industrial IoT, Big Data and Supply Chain (IIoTBDSC), pp. 317–321 (2022). IEEE

[20] Bruzzone, A.G., Cianci, R., Sciomachen, A., Sinelshchikov, K., Agresta, M.: A digital twin approach to develop a new autonous system able to operate in high temperature environments within industrial plants. In: Summer Simulation Conference, pp. 1–11 (2019)

[21] Stegmaier, V., Eberhardt, T., Schaaf, W., Jazdi, N., Weyrich, M.: A behavior model for Digital Twins of vacuum suction cups. Procedia CIRP **118**, 958–963 (2023)

[22] Lopez, V., Akundi, A.: A Conceptual Model-based Systems Engineering (MBSE) approach to develop Digital Twins. In: IEEE Int. Systems Conf. (SysCon) (2022). IEEE

[23] Otto, B., Kleinert, T.: A Flow Graph based Approach for controlled Generation of AAS Digital Twin Instances for the Verification of Compliance Check Tools. In: 48th Annual Conf. of the IEEE Industrial Electronics Society (IECON). IEEE, New York, NY, USA

(2022)

[24] Parri, J., Patara, F., Sampietro, S., Vicario, E.: A framework for Model-Driven Engineering of resilient software-controlled systems. Computing **103**(4), 589–612 (2020)

[25] Ding, K., Fan, L.-q.: AML-based web-twin visualization integration framework for DT-enabled and IIoT-driven Manufacturing system under I4.0 workshop. Journal of Manufacturing Systems **64**, 479–496 (2022)

[26] Lehner, D., Sint, S., Vierhauser, M., Narzt, W., Wimmer, M.: AML4DT: A Model-Driven Framework for Developing and Maintaining Digital Twins with AutomationML. In: 26th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA) (2021). IEEE

[27] Shi, J., Pan, Z., Jiang, L., Zhai, X.: An ontology-based methodology to establish city information model of digital twin city by merging BIM, GIS and IoT. Advanced Engineering Informatics **57**, 102114 (2023)

[28] Zake, M., Majore, G.: Application of Multi-perspective Modelling Approach for Building Digital Twin in Smart Agriculture. In: 63rd Int. Scientific Conf. on Information Technology and Management Science of Riga Technical University (ITMS) (2022). IEEE

[29] Júnior, A.O., Calvo-Rolle, J.-L., Leitão, P.: Artificial Intelligence Data-Driven Petri nets Approach for Virtualizing Digital Twins. In: IEEE Int. Conf. on Industrial Technology (ICIT). IEEE, New York, NY, USA (2023)

[30] Galkin, N., Ruchkin, M., Vyatkin, V., Yang, C.-W., Dubinin, V.: Automatic Generation of Data Centre Digital Twins for Virtual Commissioning of Their Automation Systems. IEEE Access **11**, 4633–4644 (2023)

[31] Liu, C., Wang, C., Zhang, Z., Lyu, P., Cheung, C.F.: A high-fidelity digital twin approach for the optimisation of fluid jet polishing process. Procedia CIRP **107**, 101–106 (2022)

[32] Nuy, L., Rotering, J., Rachner, J., Kiesel, R., Schmitt, R.H.: Conception of a data model for a digital twin for context-specific work instructions. Procedia CIRP **118**, 312–317 (2023)

[33] Pause, D., Blum, M.: Conceptual Design of a Digital Shadow for the Procurement of Stocked Products, pp. 288–295. Springer, Cham (2018)

[34] Jiang, L., Su, S., Pei, X., Chu, C., Yuan, Y., Wang, K.: Product-part level digital twin modeling method for digital thread framework. Computers & Industrial Engineering **179**, 109168 (2023)

[35] Brockhoff, T., Heithoff, M., Koren, I., Michael, J., Pfeiffer, J., Rumpe, B., Uysal, M.S., Van Der Aalst, W.M.P., Wortmann, A.: Process Prediction with Digital Twins. In: ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems Comp. (MODELS-C). IEEE, New York, NY, USA (2021)

[36] Sánchez, M.A., Rossit, D., Tohmé, F.: Modelling the Dynamics of a Digital Twin, pp. 238–252. Springer, Cham (2021)

[37] Bachelor, G., Brusa, E., Ferretto, D., Mitschke, A.: Model-Based Design of Complex Aeronautical Systems Through Digital Twin and Thread Concepts. IEEE Systems Journal **14**(2) (2020)

[38] Glatt, M., Sinnwell, C., Yi, L., Donohoe, S., Ravani, B., Aurich, J.C.: Modeling and implementation of a digital twin of material flows based on physics simulation. Journal of Manufacturing Systems **58**, 231–245 (2021)

[39] Harries, T., Hartnoll, M., Hafezianrazavi, M., Meek, H., Nassehi, A.: Digital Twins for Predictive Maintenance. Procedia CIRP **118**, 306–311 (2023)

[40] Chen, K., Zhu, X., Anduv, B., Jin, X., Du, Z.: Digital twins model and its updating method for heating, ventilation and air conditioning system using broad learning system algorithm. Energy **251**, 124040 (2022)

[41] Tao, F., Sun, X., Cheng, J., Zhu, Y., Liu, W., Wang, Y., Xu, H., Hu, T., Liu, X., Liu, T., Sun, Z., Xu, J., Bao, J., Xiang, F., Jin, X.: makeTwin: A reference architecture for digital twin software platform. Chinese Journal of Aeronautics **37**(1), 1–18 (2024)

[42] De Giacomo, G., Favorito, M., Leotta, F., Mecella, M., Silo, L.: Digital twin composition in smart manufacturing via Markov decision processes. Computers in Industry **149**, 103916 (2023)

[43] Petri, I., Rezgui, Y., Ghoroghi, A., Alzahrani, A.: Digital twins for performance management in the built environment. Journal of Industrial Information Integration **33**, 100445 (2023)

[44] Bardaro, G., Semprebon, A., Matteucci, M.: A use case in model-based robot development using AADL and ROS. In: 1st Int. Workshop on Robotics Software Engineering. ICSE '18. ACM, New York, NY, USA (2018)

[45] Bolender, T., Burvenich, G., Dalibor, M., Rumpe, B., Wortmann, A.: Self-Adaptive Manufacturing with Digital Twins. In: Int. Symp. on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). IEEE, New York, NY, USA (2021)

[46] Barat, S., Kulkarni, V., Bhattacharya, K.: Enterprise Digital Twins for Risk Free Business Experimentations. In: Winter Simulation Conference (WSC). IEEE, New York, NY, USA (2022)

[47] Zhang, C., Zhou, G., Jing, Y., Wang, R., Chang, F.: A Digital Twin-Based Automatic Programming Method for Adaptive Control of Manufacturing Cells. IEEE Access **10**, 80784–80793 (2022)

[48] Yu, G., Wang, Y., Mao, Z., Hu, M., Sugumaran, V., Wang, Y.K.: A digital twin-based decision analysis framework for operation and maintenance of tunnels. Tunnelling and Underground Space Technology **116**, 104125 (2021)

[49] Huang, Z., Fey, M., Liu, C., Beysel, E., Xu, X., Brecher, C.: Hybrid learning-based digital twin for manufacturing process: Modeling framework and implementation. Robotics and Computer-Integrated Manufacturing **82**, 102545 (2023)

[50] Yu, J., Liu, P., Li, Z.: Hybrid modelling and digital twin development of a steam turbine control stage for online performance monitoring. Renewable and Sustainable Energy Reviews **133**, 110077 (2020)

[51] Jin, J., Hu, J., Li, C., Shi, Z., Lei, P., Tian, W.: A Digital Twin system of reconfigurable tooling for monitoring and evaluating in aerospace assembly. Journal of Manufacturing Systems **68**, 56–71 (2023)

[52] Anbalagan, A., Shivakrishna, B., Srikanth, K.S.: A digital twin study for immediate design / redesign of impellers and blades: Part 1: CAD modelling and tool path simulation. Materials Today: Proceedings **46**, 8209–8217 (2021)

[53] Yang, Y., Meng, W., Zhu, S.: A Digital Twin Simulation Platform for Multi-rotor UAV. In: 7th Int. Conf. on Information, Cybernetics, and Computational Social Systems (ICCSS). IEEE, New York, NY, USA (2020)

[54] Yang, Y., Meng, W., Li, H., Lu, R., Fu, M.: A Digital Twin Platform for Multi-Rotor UAV. In: 40th Chinese Control Conference (CCC). IEEE, New York, NY, NY (2021)

[55] Hosamo, H.H., Svennevig, P.R., Svidt, K., Han, D., Nielsen, H.K.: A Digital Twin predictive maintenance framework of air handling units based on automatic fault detection and diagnostics. Energy and Buildings **261**, 111988 (2022)

[56] Vještica, M., Dimitrieski, V., Pisarić, M.M., Kordić, S., Ristić, S., Luković, I.: Production processes modelling within digital product manufacturing in the context of Industry 4.0. International Journal of Production Research **61**(19), 6271–6290 (2022)

[57] Llopis, J.A., Criado, J., Iribarne, L., Muñoz, P., Troya, J., Vallecillo, A.: Modeling and

Synchronizing Digital Twin Environments. In: Annual Modeling and Simulation Conference (ANNSIM), pp. 245–257 (2023). IEEE

[58] Barat, S., Kulkarni, V., Clark, T., Barn, B.: Digital twin as risk-free experimentation aid for techno-socio-economic systems. In: 25th Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), pp. 66–75 (2022)

References

[59] Javaid, M., Haleem, A.: Digital Twin applications toward Industry 4.0: A Review. Cognitive Robotics (2023)

[60] Zech, P., Clark, T., Breu, R.: An empirical analysis of digital twin adoption. In: 58th Hawaiin Int. Conf. on System Sciences (HICSS'58), pp. 6253–6262 (2025)

[61] Liu, M., Fang, S., Dong, H., Xu, C.: Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems **58**, 346–361 (2021)

[62] Erol, T., Mendi, A.F., Doğan, D.: The digital twin revolution in healthcare. In: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1–7 (2020). IEEE

[63] Deng, T., Zhang, K., Shen, Z.-J.M.: A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. Journal of Management Science and Engineering **6**(2), 125–134 (2021)

[64] Yu, W., Patros, P., Young, B., Klinac, E., Walmsley, T.G.: Energy digital twin technology for industrial energy management: Classification, challenges and future. Renewable and Sustainable Energy Reviews **161**, 112407 (2022)

[65] Tao, F., Zhang, M.: Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing. IEEE Access **5**, 20418–20427 (2017)

[66] Fitzgerald, J., Gamble, C., Mansfield, M., Ouy, J., Palacin, R., Pierce, K., Larsen, P.G.: Collaborative modelling and co-simulation for Transportation Cyber-Physical Systems. In: Transportation Cyber-Physical Systems, pp. 51–79 (2018). Elsevier

[67] Errandonea, I., Beltrán, S., Arrizabalaga, S.: Digital Twin for maintenance: A literature review. Computers in Industry **123**, 103316 (2020)

[68] Hosamo, H.H., Nielsen, H.K., Alnmr, A.N., Svennevig, P.R., Svidt, K.: A review of the Digital Twin technology for fault detection in buildings. Frontiers in Built Environment **8**, 1013196 (2022)

[69] Somers, R.J., Douthwaite, J.A., Wagg, D.J., Walkinshaw, N., Hierons, R.M.: Digital-twin-based testing for cyber–physical systems: A systematic literature review. Information and Software Technology **156**, 107145 (2023)

[70] Arowoiya, V.A., Moehler, R.C., Fang, Y.: Digital twin technology for thermal comfort and energy efficiency in buildings: A state-of-the-art and future directions. Energy and Built Environment, 2666123323000314 (2023)

[71] Ogunsakin, R., Mehandjiev, N., Marin, C.A.: Towards adaptive digital twins architecture. Computers in Industry **149**, 103920 (2023)

[72] Stary, C., Elstermann, M., Fleischmann, A., Schmidt, W.: Behavior-centered digital-twin design for dynamic cyber-physical system development. Complex Systems Informatics and Modeling Quarterly (CSIMQ) **30**, 31–52 (2022)

[73] Boschert, S., Rosen, R.: Digital Twin—The Simulation Aspect. In: Hehenberger, P., Bradley, D. (eds.) Mechatronic Futures, pp. 59–74. Springer, Cham (2016)

[74] Attaran, M., Celik, B.G.: Digital Twin: Benefits, use cases, challenges, and opportunities. Decision Analytics Journal **6**, 100165

(2023)

[75] ISO/IEC: ISO/IEC 30173: Digital twin – concepts and terminology. Technical report, International Standardization Organization (ISO) (2023)

[76] Tao, F., Zhang, M., Nee, A.Y.C.: Five-Dimension Digital Twin Modeling and Its Key Technologies. In: Tao, F., Zhang, M., Nee, A.Y.C. (eds.) Digital Twin Driven Smart Manufacturing, pp. 63–81 (2019)

[77] Boyes, H., Watson, T.: Digital twins: An analysis framework and open issues. Computers in Industry **143**, 103763 (2022)

[78] Van Den Brand, M., Cleophas, L., Gunasekaran, R., Haverkort, B., Negrin, D.A.M., Muctadir, H.M.: Models Meet Data: Challenges to Create Virtual Entities for Digital Twins. In: ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 225–228 (2021)

[79] Bordeleau, F., Combemale, B., Eramo, R., Brand, M., Wimmer, M.: 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, pp. 43–54 (2020). Springer

[80] Wymore, A.W.: Model-based Systems Engineering vol. 3. CRC press, Boca Raton, FL, USA (2018)

[81] Muñoz, P., Zschaler, S., Paige, R.F.: Preface to the special issue on success stories in model driven engineering. Science of Computer Programming **233**, 103072 (2024)

[82] Whittle, J., Hutchinson, J., Rouncefield, M.: The state of practice in model-driven engineering. IEEE software **31**(3), 79–85 (2013)

[83] Kitchenham, B.A., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, EBSE Technical Report, School of Computer Science and Mathematics, Keele University and Department of Computer Science, University of Durham, Keele, UK (July 2007). Last accessed 11 June 2025. https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf

[84] Lehner, D., Zhang, J., Pfeiffer, J., Sint, S., Splettstößer, A.-K., Wimmer, M., Wortmann, A.: Model-driven engineering for digital twins: a systematic mapping study. Software and Systems Modeling, 1–39 (2025)

[85] Sampaio, A.: Improving systematic mapping reviews. ACM SIGSOFT Software Engineering Notes **40**(6), 1–8 (2015)

[86] Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering–a systematic literature review. Information and software technology **51**(1), 7–15 (2009)

[87] Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)

[88] Paige, R.F., Cabot, J.: What makes a good modeling research contribution? Software and Systems Modeling, 1–5 (2024)

[89] Gil, S., Oakes, B., Gomes, C., Frasheri, M., Larsen, P.G.: Towards a Systematic Reporting Framework for Digital Twins: A Cooperative Robotics Case Study. SIMULATION (2024) https://doi.org/10.1177/00375497241261406

[90] Ji, T., Huang, H., Xu, X.: Digital Twin Technology — A bibliometric study of top research articles based on Local Citation Score. Journal of Manufacturing Systems **64**, 390–408 (2022)

[91] Wang, J., Li, X., Wang, P., Liu, Q.: Bibliometric analysis of digital twin literature: a review of influencing factors and conceptual structure. Technology Analysis & Strategic Management, 1–15 (2022)

[92] Estefan, J.A., *et al.*: Survey of model-based systems engineering (MBSE) methodologies. Incose MBSE Focus Group **25**(8), 1–12 (2007)

[93] Huldt, T., Stenius, I.: State-of-practice survey of model-based systems engineering. Systems engineering **22**(2), 134–145 (2019)

[94] Campo, K.X., Teper, T., Eaton, C.E., Shipman, A.M., Bhatia, G., Mesmer, B.: Model-Based Systems Engineering: Evaluating Perceived Value, Metrics, and Evidence Through Literature. Systems Engineering **26**(1), 104–129 (2023)

[95] Henderson, K., Salado, A.: Value and Benefits of Model-Based Systems Engineering (Mbse): Evidence From the Literature. Systems Engineering **24**(1), 51–66 (2021)

[96] Huldt, T., Stenius, I.: State-of-practice survey of model-based systems engineering. Systems Engineering **22**(2), 134–145 (2019)

[97] Duffy, J.B., Feng, J., Combs, R., Richardson, J.P.: Return on Investment in Model-Based Systems Engineering Software Tools. In: INCOSE International Symposium, vol. 31, pp. 791–805 (2021). Wiley Online Library

[98] (OMG), O.M.G.: Unified Modelling Language (UML) Specification, Version 2.5.1. OMG Document Number formal/17-12-05 (https://www.omg.org/spec/UML/2.5.1) (2017)

[99] (OMG), O.M.G.: OMG Systems Modeling Language (SysML) Specification, Version 1.6. OMG Document Number formal/19-11-01 (https://www.omg.org/spec/SysML/1.6) (2019)

[100] Zarour, K., Benmerzoug, D., Guermouche, N., Drira, K.: A systematic literature review on BPMN extensions. Business Process Management Journal **26**(6), 1473–1503 (2020)

[101] Lúcio, L., Mustafiz, S., Denil, J., Vangheluwe, H., Jukss, M.: FTG+ PM: An integrated framework for investigating model transformation chains. In: SDL 2013: Model-Driven Dependability Engineering: 16th International SDL Forum, Montreal, Canada, June 26-28, 2013. Proceedings 16, pp. 182–202 (2013). Springer

[102] Lankhorst, M.M., Proper, H.A., Jonkers, H.: The anatomy of the archimate language. International Journal of Information System Modeling and Design (IJISMD) **1**(1), 1–32 (2010)

[103] Lankhorst, M.M., Proper, H.A., Jonkers, H.: The architecture of the archimate language. In: International Workshop on Business Process Modeling, Development and Support, pp. 367–380 (2009). Springer

[104] Krogstie, J., Jørgensen, H.D.: Interactive models for supporting networked organisations. In: Advanced Information Systems Engineering: 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004. Proceedings 16, pp. 550–563 (2004). Springer

[105] Engels, G., Heckel, R., Sauer, S.: UML—a universal modeling language? In: Application and Theory of Petri Nets 2000: 21st International Conference, ICATPN 2000 Aarhus, Denmark, June 26–30, 2000 Proceedings 21, pp. 24–38 (2000). Springer

[106] Kruus, H., Robal, T., Jervan, G.: Teaching modeling in SysML/UML and problems encountered. In: 2014 25th EAEEIE Annual Conference (EAEEIE), pp. 33–36 (2014). IEEE

[107] Fowler, M.: Domain-specific Languages. Pearson Education, New York, NY, USA (2010)

[108] Voelter, M., Kolb, B., Birken, K., Tomassetti, F., Alff, P., Wiart, L., Wortmann, A., Nordmann, A.: Using language workbenches and domain-specific languages for safety-critical software development. Software & Systems Modeling **18**(4), 2507–2530 (2019)

[109] Bucchiarone, A., Ciccozzi, F., Lambers, L.,

Pierantonio, A., Tichy, M., Tisi, M., Wortmann, A., Zaytsev, V.: What Is the Future of Modeling? IEEE Software **38**(2), 119–127 (2021)

[110] Michael, J., Cleophas, L., Zschaler, S., Clark, T., Combemale, B., Godfrey, T., Khelladi, D., Kulkarni, V., Lehner, D., Rumpe, B., Wimmer, M., Wortmann, A., Ali, S., Barn, B., Barosan, I., Bencomo, N., Bordeleau, F., Grossmann, G., Karsai, G., Kopp, O., Mitschang, B., Muñoz Ariza, P., Pierantonio, A., Polack, F., Riebisch, M., Schlingloff, H., Stumptner, M., Vallecillo, A., van den Brand, M., Vangheluwe, H.: Model-Driven Engineering for Digital Twins: Opportunities and Challenges. INCOSE Systems Engineering (2025) https://doi.org/10.1002/sys.21815

[111] Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W.: Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine **51**(11), 1016–1022 (2018)

[112] Michael, J., Blankenbach, J., Derksen, J., Finklenburg, B., Fuentes, R., Gries, T., Hendiani, S., Herlé, S., Hesseler, S., Kimm, M., Kirchhof, J.C., Rumpe, B., Schüttrumpf, H., Walther, G.: Integrating models of civil structures in digital twins: State-of-the-Art and challenges. Journal of Infrastructure Intelligence and Resilience **3**(3) (2024) https://doi.org/10.1016/j.iintel.2024.100100

[113] Vangheluwe, H., De Lara, J., Mosterman, P.J.: An introduction to multi-paradigm modelling and simulation. In: AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems), vol. 21 (2002)

[114] Zech, P., Fröch, G., Breu, R.: A Requirements Study on Model Repositories for Digital Twins in Construction Engineering. In: Int. Conf. on Cooperative Information Systems, pp. 459–469 (2023). Springer

[115] Zech, P., Pobitzer, P., Fröch, G., Breu, R.: A Proposal for a Models-meet-Data Repository For Digital Twins in Construction Engineering. In: 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C), pp. 1–8. IEEE, New York, NY, NY (2024)

[116] Kahani, N., Bagherzadeh, M., Cordy, J.R., Dingel, J., Varró, D.: Survey and classification of model transformation tools. Software & Systems Modeling **18**(4), 2361–2397 (2019)

[117] Semeraro, C., Lezoche, M., Panetto, H., Dassisti, M.: Digital twin paradigm: A systematic literature review. Computers in Industry **130**, 103469 (2021)

[118] Bencomo, N., Götz, S., Song, H.: Models@run.time: a guided tour of the state of the art and research challenges. Software & Systems Modeling **18**(5), 3049–3082 (2019)

[119] Zalila, F., Challita, S., Merle, P.: Model-driven cloud resource management with OCCIware. Future Generation Computer Systems **99**, 260–277 (2019)

[120] Garlan, D., Schmerl, B.: Model-based adaptation for self-healing systems. In: First Workshop on Self-healing Systems, pp. 27–32 (2002)

[121] Flammini, F.: Digital twins as run-time predictive models for the resilience of cyber-physical systems: a conceptual framework. Philosophical Transactions of the Royal Society A **379**(2207), 20200369 (2021)

[122] Fritzson, P.: Principles of Object-oriented Modeling and Simulation with Modelica 3.3: a Cyber-physical Approach. John Wiley & Sons, Hoboken, NJ, USA (2014)

[123] Helbing, D.: Agent-based modeling. In: Social Self-organization: Agent-based Simulations and Experiments to Study Emergent Social Behavior, pp. 25–70. Springer, Cham (2012)

[124] Boschert, S., Rosen, R.: Digital twin—the simulation aspect. Mechatronic futures: Challenges and solutions for mechatronic systems and their designers, 59–74 (2016)

[125] Oakes, B., Gomes, C., Denil, J., Deantoni, J., Cambeiro, J., Fitzgerald, J., Larsen, P.G.: Examining Model Qualities and Their Impact on Digital Twins. In: 2023 Annual Modeling and Simulation Conference (ANNSIM), pp. 220–232. IEEE, New York, NY, NY (2023). IEEE

[126] Stachowiak, H.: Allgemeine Modelltheorie. Springer Verlag, Wien New York (1973)

[127] Mayr, H.C., Thalheim, B.: The triptych of conceptual modeling: A framework for a better understanding of conceptual modeling. Softw. Syst. Model. **20**(1), 7–24 (2021) https://doi.org/10.1007/s10270-020-00836-z

[128] Combemale, B., Kienzle, J., Mussbacher, G., Ali, H., Amyot, D., Bagherzadeh, M., Batot, E., Bencomo, N., Benni, B., Bruel, J.-M., *et al.*: A Hitchhiker's Guide to Model-Driven Engineering for Data-Centric Systems. IEEE Software **38**(4), 71–84 (2020)

[129] Barn, B.S.: The sociotechnical digital twin: On the gap between social and technical feasibility. In: IEEE 24th Conference on Business Informatics (CBI), vol. 1, pp. 11–20 (2022). IEEE

[130] Mihai, S., Yaqoob, M., Hung, D.V., Davis, W., Towakel, P., Raza, M., Karamanoglu, M., Barn, B., Shetve, D., Prasad, R.V., *et al.*: Digital twins: A survey on enabling technologies, challenges, trends and future prospects. IEEE Communications Surveys & Tutorials **24**(4), 2255–2291 (2022)

[131] Lee, E.A.: Modeling in engineering and science. Communications of the ACM **62**(1), 35–36 (2018)

[132] Gerlee, P., Lundh, T.: Scientific models. Red Atoms, White Lies and Black Boxes in a Yellow Book. Switzerland: Springer (2016)

[133] Wright, L., Davidson, S.: How to tell the difference between a model and a digital twin. Advanced Modeling and Simulation in Engineering Sciences **7**, 1–13 (2020)

[134] Cao, H., Folan, P.: Product life cycle: the evolution of a paradigm and literature review from 1950–2009. Production Planning & Control **23**(8), 641–662 (2012)

[135] Miller, R., Collins, C.T.: Acceptance testing. Proc. XPUniverse **238** (2001)

[136] Chauhan, V.K.: Smoke testing. Int. J. Sci. Res. Publ **4**(1), 2250–3153 (2014)

[137] Tao, F., Xiao, B., Qi, Q., Cheng, J., Ji, P.: Digital twin modeling. Journal of Manufacturing Systems **64**, 372–389 (2022)

[138] Felderer, M., Zech, P., Breu, R., Büchler, M., Pretschner, A.: Model-based security testing: a taxonomy and systematic classification. Software Testing, Verification and Reliability **26**(2), 119–148 (2016)

[139] Rathore, M.M., Shah, S.A., Shukla, D., Bentafat, E., Bakiras, S.: The Role of AI, Machine Learning, and Big Data in Digital Twinning: A Systematic Literature Review, Challenges, and Opportunities. IEEE Access **9**, 32030–32052 (2021)

[140] Mihai, S., Yaqoob, M., Hung, D.V., Davis, W., Towakel, P., Raza, M., Karamanoglu, M., Barn, B., Shetve, D., Prasad, R.V., Venkataraman, H., Trestian, R., Nguyen, H.X.: Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. IEEE Communications Surveys & Tutorials **24**(4), 2255–2291 (2022)

[141] Fuller, A., Fan, Z., Day, C., Barlow, C.: Digital Twin: Enabling Technologies, Challenges and Open Research. IEEE Access **8**, 108952–108971 (2020)

[142] Semeraro, C., Lezoche, M., Panetto, H., Dassisti, M.: Digital twin paradigm: A systematic literature review. Computers in Industry **130**, 103469 (2021)

[143] Barricelli, B.R., Casiraghi, E., Fogli, D.: A survey on digital twin: Definitions, characteristics, applications, and design implications. IEEE access **7**, 167653–167671 (2019)

[144] Gu, X., Duan, W., Zhang, G., Hou, J., Peng, L., Wen, M., Gao, F., Chen, M., Ho, P.-H.: Digital twin technology for intelligent vehicles and transportation systems: A survey on applications, challenges and future directions. IEEE Communications Surveys & Tutorials (2025)

[145] Ghansah Dr, F.A., Udeaja, C., Darko, A., Lu, W., Agyekum-Mensah, G., Antwi-Afari, P., Twum-Ampofo, S., Ababio Mr, B.K., Debrah, C., *et al.*: Prioritising critical barriers to digital twins implementation for sustainable smart facility management in the built environment: A global study. In: CIB Conferences, vol. 1, p. 159 (2025)

[146] Katsoulakis, E., Wang, Q., Wu, H., Shahriyari, L., Fletcher, R., Liu, J., Achenie, L., Liu, H., Jackson, P., Xiao, Y., *et al.*: Digital twins for health: a scoping review. NPJ digital medicine **7**(1), 77 (2024)

[147] Kabir, M.R., Halder, D., Ray, S.: Digital twins for iot-driven energy systems: A survey. IEEE Access (2024)

[148] Bühler, L., Schuster, T., Pflaum, A.: Twin transformation - the case of smart circular economy in mmanufacturing industries: Insights from an umbrella review. International Journal of Innovation Management **29**(05n06), 2540007 (2025) https://doi.org/10.1142/S1363919625400079

[149] Dalibor, M., Jansen, N., Rumpe, B., Schmalzing, D., Wachtmeister, L., Wimmer, M., Wortmann, A.: A cross-domain systematic mapping study on software engineering for digital twins. Journal of Systems and Software **193**, 111361 (2022)

[150] Gil, S., Mikkelsen, P.H., Gomes, C., Larsen, P.G.: Survey on open-source digital twin frameworks–A case study approach. Software: Practice and Experience (2024)

[151] Automation systems and integration — Digital twin framework for manufacturing. Standard, International Organization for Standardization, Geneva, CH (March 2021)

[152] Oakes, B.J., Parsai., A., Mierlo., S.V., Demeyer., S., Denil., J., Meulenaere., P.D., Vangheluwe., H.: Improving Digital Twin Experience Reports. In: 9th Int. Conf. on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD,, pp. 179–190. SciTePress, Setúbal, Portugal (2021). INSTICC

[153] Autiosalo, J., Vepsäläinen, J., Viitala, R., Tammi, K.: A feature-based framework for structuring industrial digital twins. IEEE access **8**, 1193–1208 (2019)

[154] Newrzella, S.R., Franklin, D.W., Haider, S.: 5-dimension cross-industry digital twin applications model and analysis of digital twin classification terms and models. IEEE Access **9**, 131306–131321 (2021)

[155] Uhlenkamp, J.-F., Hauge, J.B., Broda, E., Luetjen, M., Freitag, M., Thoben, K.-D.: Digital Twins: A Maturity Model for Their Classification and Evaluation. IEEE ACCESS **10**, 69605–69635 (2022)

[156] Zambrano, V., Mueller-Roemer, J., Sandberg, M., Talasila, P., Zanin, D., Larsen, P.G., Loeschner, E., Thronicke, W., Pietraroia, D., Landolfi, G., *et al.*: Industrial digitalization in the industry 4.0 era: Classification, reuse and authoring of digital models on Digital Twin platforms. Array **14**, 100176 (2022)

[157] Hakiri, A., Gokhale, A., Yahia, S.B., Mellouli, N.: A comprehensive survey on digital twin for future networks and emerging Internet of Things industry. Computer Networks, 110350 (2024)

[158] Usman, M., Britto, R., Börstler, J., Mendes, E.: Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method. Information and Software Technology **85**, 43–59 (2017)

[159] Feng, H., Gomes, C., Thule, C., Lausdahl, K., Sandberg, M., Larsen, P.G.: The incubator case study for digital twin engineering. arXiv preprint arXiv:2102.10390 (2021)

[160] Fitzgerald, J., Gomes, C., Larsen, P.G.: The Engineering of Digital Twins. Springer, Cham (2024)

[161] Gomes, C., Kristensen, M.H., Andersen, M.S., Talasila, P., Feng, H., Wright, T., Larsen, P.G.: Digital twin tutorial: The incubator case study. In: Engineering Trustworthy Software Systems: 6th International School (SETSS 2024), Tutorial Lectures, pp. 68–101. Springer, Singapore (2025). Code for the incubator DT is found at https://github.com/INTO-CPS-Association/example_digital-twin_incubator

[162] Lana, C.A., Guessi, M., Antonino, P.O., Rombach, D., Nakagawa, E.Y.: A Systematic Identification of Formal and Semi-Formal Languages and Techniques for Software-Intensive Systems-Of-Systems Requirements Modeling. IEEE systems journal **13**(3), 2201–2212 (2018)

[163] Ozkaya, M.: Do the Informal & Formal Software Modeling Notations Satisfy Practitioners for Software Architecture Modeling? Information and Software Technology **95**, 15–33 (2018)

[164] Morand, B.: Modeling: Is It Turning Informal Into Formal? In: International Conference on the Unified Modeling Language, pp. 37–48 (1998). Springer

[165] Häring, I.: Semi-Formal Modeling of Multi-technological Systems I: UML. In: Technical Safety, Reliability and Resilience: Methods and Processes, pp. 227–263. Springer, ??? (2021)

[166] Häring, I.: Semi-formal Modeling of Multi-technological Systems II: SysML Beyond the Requirements Diagram. In: Technical Safety, Reliability and Resilience: Methods and Processes, pp. 265–275. Springer, ??? (2021)

[167] Talasila, P., Gomes, C., Vosteen, L.B., Iven, H., Leucker, M., Gil, S., Mikkelsen, P.H., Kamburjan, E., Larsen, P.G.: Composable digital twins on digital twin as a service platform. Simulation **101**(3), 287–311 (2025)

[168] Zech, P., Nardin, C., Ristov, S., Flora, M., Breu, R.: Digital-Twins-as-a-Service in Construction Engineering. In: 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), pp. 3004–3010 (2024). IEEE

[169] Harzing, A.-W.: The Publish or Perish Book. Tarma Software Research Pty Limited Melbourne, Melbourne, Australia (2010)

[170] Zech, P., Barat, S., Nast, B., Oakes, B., Michael, J., Zschaler, S., Barn, B., Breu, R.: Dataset for "Model-based Digital Twin Engineering: Insights, Challenges, and Future Directions". figshare (2024). https://doi.org/10.6084/m9.figshare.27312138

[171] Shao, G.: Manufacturing digital twin standards. In: 1st Int. Conf. on Engineering Digital Twins, pp. 1–8 (2024). ACM

[172] Visser, W., Havelund, K., Brat, G., Park, S., Lerda, F.: Model checking programs. In: Automated Software Engineering, pp. 3–12. Springer, Netherlands (2003). https://doi.org/10.1023/A:1022920129859

[173] Askarpour, M., Lestingi, L., Longoni, S., Iannacci, N.: Formally-based model-driven development of collaborative robotic applications. Journal of Intelligent & Robotic Systems **102**(2), 15–36 (2021) https://doi.org/10.1007/s10846-021-01386-2

[174] Belle, A., Shan, J., Sivakumar, M.: Prompting gpt-4 to support automatic safety case generation. Expert Systems with Applications (2024) https://doi.org/10.1016/j.eswa.2024.121987

[175] Maro, S., Anjorin, A., Wohlrab, R.: Traceability maintenance: Factors and guidelines. In: Int. Conf. on Automated Software Engineering, pp. 208–219. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2970276.2970314

[176] Muñoz, P., Wimmer, M., Troya, J., Vallecill, A.: Measuring the fidelity of a physical and a digital twin using trace

alignments. IEEE Transactions on Software Engineering (2024) https://doi.org/10.1109/TSE.2024.3462978

[177] Zech, P., Hammes, S., Moroder-Geisler, D., Hauer, M., Pfluger, R., Breu, R.: Quantifying the simulation-reality-gap in building simulation. In: Building Simulation 2025: 19th Conference of IBPSA. Building Simulation. IBPSA, USA (2025)

[178] Rosik, J., Le Gear, A., Buckley, J., Babar, M.A., Connolly, D.: Assessing Architectural Drift in Commercial Software Development: A Case Study. Software: Practice and Experience **41**(1), 63–86 (2011)

[179] Villar, E., Merino, J., Posadas, H., Henia, R., Rioux, L.: Mega-modeling of complex, distributed, heterogeneous cps systems. Microprocessors and microsystems **78**, 103244 (2020)

[180] Wu, S., Wang, G., Lu, J., Huang, J., Qiao, J., Yan, Y., Kiritsis, D.: Cognitive Digital Thread Rool-chain for Model Versioning in Model-based Systems Engineering. Advanced Engineering Informatics **67**, 103490 (2025)

[181] Adesanya, F., Silva, K.C., Neto, V.V.G., David, I.: Systems of Twinned Systems: A Systematic Literature Review. arXiv preprint arXiv:2505.19916 (2025)

[182] Bellavista, P., Bicocchi, N., Fogli, M., Giannelli, C., Mamei, M., Picone, M.: Odte: A metric for digital twin entanglement. IEEE Open Journal of the Communications Society (2024)

[183] Bonetti, F., Bucchiarone, A., Michael, J., Cicchetti, A., Marconi, A., Rumpe, B.: Digital Twins of Socio-Technical Ecosystems to Drive Societal Change. In: MODELS Companion '24: Int. Conf. on Model Driven Engineering Languages and Systems, pp. 275–286. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3686248

[184] Wohlin, C., Rainer, A.: Is it a case study?—a critical analysis and guidance. Journal of Systems and Software **192**, 111395 (2022)

[185] Combemale, B., Kienzle, J., Mussbacher, G., Archambault, P., Bruel, J.-M., Burgueño, L., Cheng, B., Cleophas, L., Engels, G., Foures, D., Klikovits, S., Kulkarni, V., Michael, J., Mosser, S., Sahraoui, H., Syriani, E., Wortmann, A.: On the Challenges of Integrating Digital Twins. In: Int. Conf. on Engineering Digital Twins (EDTconf'25). IEEE, New York, NY, USA (2025)

[186] Pfeiffer, J., Zhang, J., Combemale, B., Michael, J., Rumpe, B., Wimmer, M., Wortmann, A.: Towards a Unifying Reference Model for Digital Twins of Cyber-Physical Systems. In: Int. Conf. on Emerging Technologies and Factory Automation (ETFA'25). IEEE, New York, NY, USA (2025)

[187] Muctadir, H.M., Manrique Negrin, D.A., Gunasekaran, R., Cleophas, L., van den Brand, M., Haverkort, B.R.: Current trends in digital twin development, maintenance, and operation: an interview study. Software and Systems Modeling **23**(5), 1275–1305 (2024) https://doi.org/10.1007/s10270-024-01167-z

[188] Friedenthal, S., Moore, A., Steiner, R.: A Practical Guide to SysML: The Systems Modeling Language, 3rd edn. Morgan Kaufmann, Burlington, MA, USA (2014)

[189] Weilkiens, T.: Systems Engineering with SysML/UML: Modeling, Analysis, Design. Elsevier, Burlington, MA, USA (2011)

[190] MODELISAR consortium and Modelica Association Project "FMI": Functional Mockup Interface standard 3.0. https://fmi-standard.org/docs/3.0/ (2024)

[191] Bader, S.R., Maleshkova, M.: The Semantic Asset Administration Shell. In: Semantic Systems. The Power of AI and Knowledge Graphs: 15th International Conference, SEMANTiCS 2019., pp. 159–174. Springer, ??? (2019)

[192] Zhang, J., Ellwein, C., Heithoff, M.,

Michael, J., Wortmann, A.: Digital Twin and the Asset Administration Shell: An Analysis of 3 AASs Types and their Feasibility for Digital Twin Engineering. Journal Software and Systems Modeling (SoSyM) **24** (2025) https://doi.org/10.1007/s10270-024-01255-0

[193] Marchezan, L., Homolka, M., Blokhin, A., Assunção, W.K.G., Herac, E., Egyed, A.: A tool for collaborative consistency checking during modeling. In: ACM/IEEE 27th Int. Conf. on Model Driven Engineering Languages and Systems. MODELS Companion '24, pp. 655–659. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3688558

[194] Raţiu, C.C., Assunção, W.K.G., Haas, R., Egyed, A.: Reactive links across multi-domain engineering models. In: 25th Int. Conf. on Model Driven Engineering Languages and Systems. MODELS '22, pp. 76–86. ACM, New York, NY, USA (2022). https://doi.org/10.1145/3550355.3552446

[195] Raţiu, C.-C., Assunção, W.K.G., Herac, E., Haas, R., Lauwerys, C., Egyed, A.: Using reactive links to propagate changes across engineering models. Software and Systems Modeling (2024) https://doi.org/10.1007/s10270-024-01186-w

[196] Bucchiarone, A., Combemale, B., Pierantonio, A., Bencomo, N., van den Brand, M., Bruel, J.-M., Cicchetti, A., Di Rocco, J., Lambers, L., Michael, J., Rumpe, B., Sjödin, M., Taentzer, G., Tichy, M., Vangheluwe, H., Wimmer, M., Zschaler, S.: Modeling: The Heart and Soul of Engineering Smart Ecosystems. In: Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS-C): SAM Conference. ACM/IEEE, New York, NY, USA (2025)

[197] Microsoft Azure IoT Hub. https://azure.microsoft.com/services/iot-hub/. [Online; accessed 05-December-2023] (2023)

[198] Microsoft Azure Digital Twins. https://azure.microsoft.com/services/digital-twins/. [Online; accessed 05-December-2023] (2023)

[199] AWS IoT Greengrass. https://aws.amazon.com/de/greengrass/. [Online; accessed 05-December-2023] (2023)

[200] AWS TwinMaker. https://aws.amazon.com/de/iot-twinmaker/. [Online; accessed 05-December-2023] (2023)

[201] Siemens Mind-Sphere. https://siemens.mindsphere.io/en. [Online; accessed 05-December-2023] (2023)

[202] Alskaif, T., Babur, Ö., Bordeleau, F., Cleophas, L., Combemale, B., Denil, J., Haugen, O.y., Michael, J., Nguyen, P., Seceleanu, T., van den Brand, M., Vangheluwe, H.: Evolution at the Core of Digital Twin Engineering. In: Int. Conf. on Engineering Digital Twins (EDTconf'25). IEEE, New York, NY, USA (2025)

[203] Mertens, J., Denil, J.: Digital-twin co-evolution using continuous validation. In: ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023). ICCPS '23, pp. 266–267. ACM, New York, NY, USA (2023). https://doi.org/10.1145/3576841.3589628

[204] Zech, P., Fröch, G., Breu, R.: A Requirements Study on Model Repositories for Digital Twins in Construction Engineering. In: International Conference on Cooperative Information Systems, pp. 459–469 (2023). Springer

[205] Hua, E.Y., Lazarova-Molnar, S., Francis, D.P.: Validation of digital twins: Challenges and opportunities. In: 2022 Winter Simulation Conference (WSC), pp. 2900–2911 (2022). https://doi.org/10.1109/WSC57314.2022.10015420

[206] Fouquet, F., Hartmann, T., Cecchinel, C., Combemale, B.: Greycat: A framework to develop digital twins at large scale. In: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. MODELS Companion '24, pp. 492–495. Association for Computing

Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3688265

[207] Fedeli, A., Manrique Negrin, D.A.: Towards a collaborative approach for digital twin simulation models comprehension. In: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. MODELS Companion '24, pp. 660–664. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3688559

[208] Ben Chaaben, M., Ben Sghaier, O., Dhaouadi, M., Elrasheed, N., Darif, I., Jaoua, I., Oakes, B., Syriani, E., Hamdaqa, M.: Toward intelligent generation of tailored graphical concrete syntax. In: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. MODELS '24, pp. 160–171. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3640310.3674085

[209] De Carlo, G., Langer, P., Bork, D.: Advanced visualization and interaction in glsp-based web modeling: realizing semantic zoom and off-screen elements. In: 25th International Conference on Model Driven Engineering Languages and Systems. MODELS '22, pp. 221–231. ACM, New York, NY, USA (2022). https://doi.org/10.1145/3550355.3552412

[210] Ciccozzi, F., Tichy, M., Vangheluwe, H., Weyns, D.: Blended modelling: what, why and how. In: 22nd Int. Conf. on Model Driven Engineering Languages and Systems Companion. MODELS '19 Companion, pp. 425–430. IEEE Press, ??? (2021). https://doi.org/10.1109/MODELS-C.2019.00068

[211] Vaquero-Melchor, D., Palomares, J., Guerra, E., Lara, J.: Active domain-specific languages: Making every mobile user a modeller. In: ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), pp. 75–82 (2017). https://doi.org/10.1109/MODELS.2017.13

[212] Archambault, P., Sahraoui, H., Syriani, E.: A modeling methodology for crop representation in digital twins for smart farming. In: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. MODELS Companion '24, pp. 342–352. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3688247

[213] Fiter, K., Malassigné-Onfroy, L., Oakes, B.: DTInsight: A tool for explicit, interactive, and continuous digital twin reporting. In: 2025 ACM/IEEE 28th International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 139–143 (2025). https://doi.org/10.1109/MODELS-C68889.2025.00030

[214] Zech, P., Senoner, S., Goldin, E., Zallinger, C., Hammes, S., Michael, J.: Model-driven Digital Twins for AECO. In: Int. Conf. on Engineering Digital Twins (EDTconf'25). IEEE, New York, NY, USA (2025)

[215] Hermann, A.-K., König, L., Burger, E., Reussner, R.: Towards integrating low-code in view-based development. In: ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems. MODELS Companion '24, pp. 866–875. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3688333

[216] Alfonso, I., Conrardy, A., Sulejmani, A., Nirumand, A., Ul Haq, F., Gomez-Vazquez, M., Sottet, J.-S., Cabot, J.: Building besser: An open-source low-code platform. In: Aa, H., Bork, D., Schmidt, R., Sturm, A. (eds.) Enterprise, Business-Process and Information Systems Modeling, pp. 203–212. Springer, Cham (2024)

[217] Martínez-Lasaca, F., Díez, P., Guerra, E., Lara, J.: A model and workflow-driven approach for engineering domain-specific low-code platforms and applications. Software and Systems Modeling (2025) https://doi.org/10.1007/s10270-025-01308-y

[218] Giraudet, T.: Interactivity and collaboration in the context of heterogeneous modeling. In: ACM/IEEE 27th Int. Conf. on Model Driven Engineering Languages and

Systems. MODELS Companion '24, pp. 174–179. ACM, New York, NY, USA (2024). https://doi.org/10.1145/3652620.3688205

[219] Franzago, M., Ruscio, D.D., Malavolta, I., Muccini, H.: Collaborative model-driven software engineering: A classification framework and a research map. IEEE Transactions on Software Engineering **44**(12), 1146–1175 (2018) https://doi.org/10.1109/TSE.2017.2755039

[220] Hopmann, M., Gutierrez, M., Thalmann, D., Vexo, F.: Bridging Digital and Physical Worlds Using Tangible Drag-and-Drop Interfaces, pp. 1–18. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22336-5_1

[221] Fett, M., Wilking, F., Goetz, S., Wartzack, S., Kirchner, E.: Practical insights into the perception of digital twins: An analysis of surveys. Forschung im Ingenieurwesen **89**(1) (2025) https://doi.org/10.1007/s10010-025-00879-y

[222] Rudolph, M., Kurz, S., Rakitsch, B.: Hybrid modeling design patterns. Journal of Mathematics in Industry **14**(1) (2024) https://doi.org/10.1186/s13362-024-00141-0

[223] Combemale, B., Gray, J., Rumpe, B.: Model hybridization: towards a unifying theory for inductive and deductive reasoning. Software and Systems Modeling **23**(6), 1307–1308 (2024) https://doi.org/10.1007/s10270-024-01249-y

[224] Zech, P., Goldin, E., Hammes, S., Moroder, D.-G., Pfluger, R., Breu, R.: Model-based auto-commissioning of building control systems. In: 26th Int. Conf. on Enterprise Information Systems - Volume 2: ICEIS, pp. 121–128 (2024). SciTePress

[225] Kautz, O., Roth, A., Rumpe, B.: In: Gruhn, V., Striemer, R. (eds.) Achievements, Failures, and the Future of Model-Based Software Engineering, pp. 221–236. Springer, Cham (2018)

[226] Olivera, P., Oquendo, F., Tekinerdogan, B., Zech, P., Lüken-Winkels, B., Barat, S., Kulkarni, V.: International Workshop on Digital Twin Architecture (TwinArch) and Digital Twin Engineering (DTE). Online: https://www.iese.fraunhofer.de/en/events/twinarch-dte.html, last accessed June 13, 2024 (2024)

[227] Barat, S., Paranjape, A.: Workshop on Digital Twin Engineering. Online: https://isec23.iiita.ac.in/workshop2.php, last accessed July 28, 2023 (2023)

[228] Cleophas, L., Michael, J., Wortmann, A.: 1st Int. Conf. on Engineering Digital Twins (EDTconf 2024). Online: https://conf.researchr.org/home/edtconf-2024, last accessed November 22, 2024 (2024)

[229] Cleophas, L., Godfrey, T., Khelladi, D.E., Lehner, D., Combemale, B., Rumpe, B., Zschaler, S.: Model-Driven Engineering of Digital Twins (Dagstuhl Seminar 22362). In: Dagstuhl Reports, vol. 12 (2023). Schloss Dagstuhl-Leibniz-Zentrum für Informatik

[230] Zech, P., Vierhauser, M., Breu, R., Vangheluwe, H., Sandkuhl, K., Barat, S., Kulkarni, V.: Interntional Workshop on Models for Simulation (M4S). Online: https://m4s.uibk.ac.at, last accessed July 10, 2024 (2024)