A Dual-Lens Conceptual Framework for Digital Twin V&V

Benoit Combemale
benoit.combemale@irisa.fr
Inria
Rennes, France

Steffen Zschaler
szschaler@acm.org
King's College London
London, United Kingdom

Jean-Michel Bruel
jean-michel.bruel@irit.fr
CNRS IRIT
Toulouse, France

Christopher Burr
cburr@turing.ac.uk
The Alan Turing Institute
London, United Kingdom

Adelaide De Vecchi
adelaide.de_vecchi@kcl.ac.uk
King's College London
London, United Kingdom

Aidan Dures
aidan.dures@kcl.ac.uk
King's College London
London, United Kingdom

Sophie Ebersold
sophie.ebersold@irit.fr
CNRS IRIT
Toulouse, France

Rym M'Hallah
rym.mhallah@kcl.ac.uk
King's College London
London, United Kingdom

Pascale Vicat-Blanc
pascale.vicatblanc@inria.fr
Inria
Lyon, France

Bofan Zhang
bofan.zhang@kcl.ac.uk
King's College London
London, United Kingdom

Abstract

Digital twins (DTs) are increasingly built and used to monitor, simulate, and improve real world systems across engineering, industry, healthcare, and urban domains. Yet, the complexity of their validation and verification (V&V) remains poorly understood. We organise the V&V challenges of DTs using a dual-lens conceptual framework that distinguishes between DTs as complex software systems and as representations of real-world systems. For each perspective, key properties are verified and validated across three dimensions —functional, temporal, and compositional— highlighting the complementarity of the two viewpoints. We then discuss future research directions for V&V of DTs.

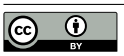
CCS Concepts

• **Software and its engineering → Abstraction, modeling and modularity**; **Software verification and validation**.

Keywords

Digital Twin, Verification and Validation

ACM Reference Format:
Benoit Combemale, Steffen Zschaler, Jean-Michel Bruel, Christopher Burr, Adelaide De Vecchi, Aidan Dures, Sophie Ebersold, Rym M'Hallah, Pascale Vicat-Blanc, and Bofan Zhang. 2026. A Dual-Lens Conceptual Framework for Digital Twin V&V. In *ACM/IEEE 29th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion 2026), October 04–09, 2026, Málaga, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3837062.3839091



MODELS Companion 2026, Málaga, Spain

ACM ISBN 979-8-4007-2903-4/2026/10
https://doi.org/10.1145/3837062.3839091

1 Introduction

Managing system adaptivity has significantly evolved over the past decades, making monitoring, analysis, and planning of system updates explicit elements of complex systems. This evolution began with Self-Adaptive Systems (SASs) [6, 8], enabling dynamic adaptation to changing conditions and uncertainty through explicit feedback loops (e.g., MAPE-K [15]). SASs enable optimization, resilience, and fault tolerance through advanced adaptive decision-making mechanisms. This makes them well-suited to complex and dynamic environments, ensuring continuous evolution and improved system performance. Digital Twins (DTs) [9, 10] introduce a major shift in this domain by fundamentally decoupling high-level control and planning from the controlled system itself. A DT is a software system that intentionally represents an original, continuously reflects its state and evolution, can influence its behavior, and delivers value to users or other systems. Unlike traditional SASs, which tightly integrate control logic with the system, DTs externalize control into a separate, software entity, enabling advanced analytics and targeted interventions to improve key performance indicators, such as sustainability, resilience, and resource efficiency [23].

Explicit separation of control from the controlled system is pivotal: control can now evolve independently of the physical system it governs. This enables more powerful and flexible control strategies, supports multiple and potentially evolving objectives, and allows control systems to be updated, replaced, or repurposed without directly modifying the original itself. Different control systems can coexist, each serving distinct goals or stakeholders.

However, this decoupling brings four significant challenges. First, ensuring that the DT maintains an accurate and timely representation of the original requires rigorous reasoning about the properties of their connection. Second, multiple control systems may attempt to steer the same original, potentially leading to conflicting objectives or unintended interactions, and requiring orchestration.

Third, the independent evolution of the control system and the original require synchronization, consistency, and long-term alignment. Fourth, DTs deployed to inform maintenance decisions, control strategies, or safety-critical operations require dependable analyses, predictions, and recommendations. These four challenges place verification and validation (V&V)—"building the right system" (validation) vs "building the system right" (verification) [13]—at the center of DT development and deployment. However, V&V of DTs remains understudied, as concluded *inter alia* by a systematic literature review of V&V for DTs for manufacturing applications [3].

This paper aims to address this gap by organising V&V concerns for DTs into a systematic conceptual framework. The framework views DTs through two complementary lenses: (i) as *complex software systems* integrating heterogeneous components (e.g., data pipelines, computational models, ML algorithms, IoT sensors and ICT networks) requiring standard V&V for standard properties such as computational correctness, reliability, performance, usability, and security; and; and (ii) as a *medium for interacting with a specific original system* that must remain aligned with the evolving state of the original system it represents and provide robust control, emphasizing issues of representational fidelity, robust and secure synchronization, instantiation, and contextual validity.

Through these two lenses, we can distinguish between the intrinsic correctness of the digital artifact, its representational fidelity, and its operational coupling with reality—identifying sources of potential DT failure. Addressing the correctness of (i) of individual components—such as models, algorithms, and data pipelines—, and; and (ii) of their integration and their evolving relationship with the original system is, thus, essential to improving DT's reliability and trustworthiness. Key DT-specific V&V research questions relate to the continuous evaluation of the validated properties, the decomposition of V&V efforts to minimise re-evaluation when either the physical or the virtual twin evolves, and the co-evolution of both the system's and of the DT's V&V.

Section 2 reviews the literature. Section 3 proposes the conceptual framework. Finally, Section 4 lists the research challenges.

2 Related work

Verification, validation, and uncertainty quantification (VVUQ) are generally recognized as "essential for the responsible development, implementation, monitoring, and sustainability of DTs" [7]. Bitencourt et al. [3] highlight the lack of consensus even on the *definition* of V&V for DTs, and identify the absence of an accepted, standardised approach for conducting V&V activities as a central gap.

To structure the V&V process for DTs in manufacturing, Bitencourt et al. [3] propose an adaptation of the V-framework for DTs, combining simulation, lifecycle, systems, and software V&V. Perisic and Perisic [20] present a more generic V&V framework based on the meta-object facility abstraction layers (instance, model, metamodel, and meta-meta-model) mapped over five helices. Sel et al. [22] examine the applications of VVUQ processes for DTs in precision medicine, with examples in cardiology and oncology, and argue that validation in DTs extends beyond traditional metrics, requiring adaptive benchmarks that reflect patient variability, ageing, and clinical objectives. They argue that VVUQ processes must maintain data security, protect individual privacy, and address ethical concerns to ensure the accuracy and integrity of DT predictions.

Despite an emerging awareness of the importance of the V&V process for DTs, a systematic analysis of its specificities is lacking, making systematic V&V of DTs difficult to achieve. Our conceptual framework is an essential first step towards addressing this gap.

3 A Conceptual Framework for DT's V&V

We detail our conceptual framework, mapping *what* needs V&V for DTs. As Figure 1 illustrates, for each of our two lenses, we distinguish three V&V dimensions: (1) the *functional* dimension highlights the critical role of a DTs' purpose (understand, analyze, anticipate, predict, support decision, control); (2) the *temporal* dimension stresses the constantly evolving nature of both DTs and their real-world counterparts; and (3) the *compositional* dimension emphasises DTs' complex internal and external composition, including model hybridation, instantiation, federation, and potentially the association of multiple DTs to a same original. Applying both lenses across these three dimensions organises the four challenges of Section 1 into a structured space of V&V concerns, from which we derive the research challenges of Section 4.

Lens 1: A Complex Software System

Verifying and validating DTs as complex software systems focuses on the *purpose of the services the DTs deliver* as determined by their intended users. This is analysed across the three dimensions.

Functional Dimension *Validation* asks whether the services are appropriate for the user to achieve their goals: are they the right services and are information presented appropriately to support robust and trustworthy decision-making/automated control [12].

Verification asks if these services are implemented correctly: Do the functions satisfy the specifications? Are predictive models working correctly given known input–output relations? Is the system performant, secure, safe, reliable, maintainable, and robust? Verification techniques comprise formal verification, testing, static analysis, and runtime monitoring. Many DTs include high-fidelity simulation models and AI components, which involve numerically intensive computations and non-determinism, making exhaustive testing and verification by strict input–output comparison infeasible. Consequently, verification must rely on selective testing, surrogate modeling, or uncertainty quantification. These limitations introduce epistemic uncertainty regarding the model's predictive capability outside tested scenarios.

Temporal Dimension *Validation* must consider the evolving users' needs, DT purpose, and operational context. A DT initially deployed for monitoring may later be used for prediction, optimization, control, or decision support. Each of these imposes different requirements on accuracy, temporal resolution, and robustness. A DT that is valid for descriptive or diagnostic purposes may not be sufficiently accurate or reliable for predictive or prescriptive applications. Contextual validity or 'fitness for purpose' must be continuously reassessed as the role of the DT expands or shifts [17].

Verification must consider the evolving nature of the (software) system itself, including changes in code, data pipelines, infrastructure technology, computational and sensor hardware. When these

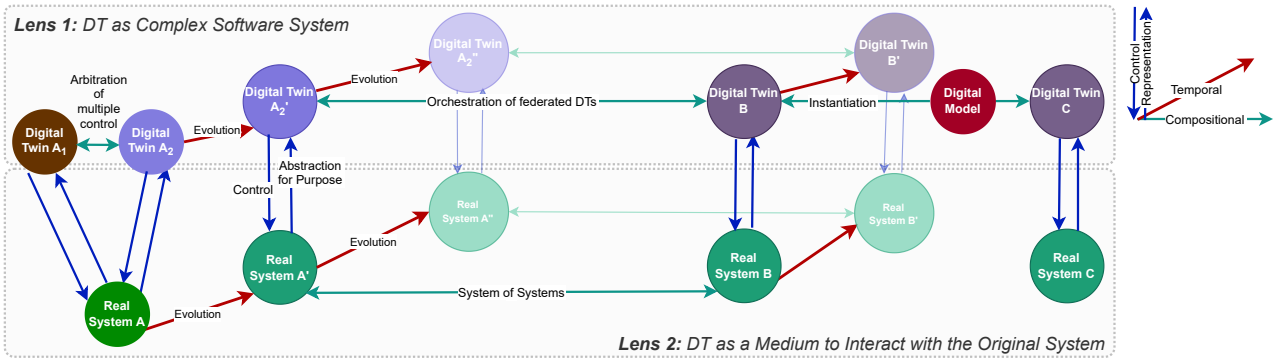


Figure 1: Conceptual framework of DT V&V

components change, they may introduce new behaviours or vulnerabilities, even if the original system was correctly implemented and verified. Therefore, verification must be an ongoing process triggered by changes in the software system.

Compositional Dimension *Validation* must explicitly consider at least three types of composition in DTs: (i) *federation* of DTs representing different original systems that interact with each other as part of a larger system of systems (e.g., a DT of a car – a federation of DTs of various components within the car – interacting with a DT of the road infrastructure) requires validation of the overall fitness-for-purpose of the federation, as well as the prevention of purpose creep when new DTs are added to the federation [16]; (ii) *arbitration* of DTs representing the same original system but with different purposes (e.g., a DT for predictive maintenance and a DT for real-time control of the same physical system) is required to ensure consistent control and decision support; and (iii) *hybridization* of mechanistic and data-driven models within a single DT (e.g., a physics-based model with a machine learning model for residual learning [21]) requires validation that training data for data-driven models match assumptions and parameters of mechanistic models.

Verification must consider DTs' complex *internal composition*. DTs often integrate heterogeneous components, e.g., data pipelines, computational models, ML algorithms, IoT sensors, and ICT networks. This internal composition introduces verification concerns related to cybersecurity, data integrity, and privacy. Security vulnerabilities may compromise DT fidelity, leading to incorrect outputs or malicious manipulation. Thus, V&V must ensure the robustness of data pipelines, communication channels, and authentication mechanisms. Tightly coupled internal components make it insufficient to verify components in isolation, and call for holistic verification approaches that address both individual components and their interactions. *DT federation* exacerbates the compositional dimension challenges, while inheriting traditional distributed systems V&V difficulties. Testing a DT federation is more complex than testing a single DT. The combination of formal verification, model checking, fault injection, unit tests, canaries, and more, can increase users' confidence in the correctness of federated DT systems, analogous to V&V in distributed systems. Where DTs use *hybrid models*, verification must face the compounded uncertainties arising from the integration of mechanistic and data-driven models. For example, a data-driven model calibrating a parameter of a mechanistic model can push the physics model outside its validity frame via a biased estimate, so that an accurate-looking fit conceals a structural error only surfacing in untested regimes. Thus, mechanistic models must be correctly calibrated and data-driven models trained following ML best practices, including cross-validation, regularization, and bias mitigation; yet neither is sufficient in isolation.

Lens 2: A Medium to Interact with the Original

Their role as representational and operational interfaces for interacting with a specific real-world system distinguishes DTs from simulation models or stand-alone software, whose value derives from internal functionality. In contrast, a DT's value stems from its ability to accurately represent, monitor, and support ongoing interaction with its physical origin. Within this lens, V&V focuses on the *relationship between the virtual model and the real system.*

Functional Dimension *Validation,* asks whether the models in the DT are fit for purpose. A (deductive) model is valid for prediction only within a specific range of assumptions, operating conditions, and parameter regimes (its *validity frame* [1]). We, therefore, must ask: Do they make appropriate assumptions? Do they capture the right information about the real system? Do they embed appropriate deductive/inductive models and their hybridisations? Are they updated at an appropriate frequency and granularity? The correctness of the DT must be evaluated not only in terms of predictive accuracy but also in terms of its impact on operational decisions. Incorrect predictions may propagate into incorrect decisions, potentially causing system degradation or safety hazards. Even correct predictions may cause issues if they ignore aspects that would materially change the decision. For example, correctly predicting sunshine hours may be insufficient for making optimal configuration decisions for a solar system without a degree of prediction of tenant behaviour. If tenant-behaviour predictions are coarse, high-resolution sunshine forecasts add no decision value and may mislead through false precision. Validation must therefore consider the DT within its operational context, including human interaction and interfaces, automated control loops, and decision-support workflows, both to ensure its proper implementation and integration and to assess its suitability for its intended use.

Verification, conversely, asks whether the models and data pipelines in the DT are implemented correctly and securely. Is the

data pipeline secure, or can data be manipulated in transfer between DT and real system? Are simulations and other calculations implemented correctly? Is uncertainty quantified and propagated appropriately? Are control actions safe?

Temporal Dimension *Validation* must consider the evolving nature of the original system and its environment. System behavior may drift over time. When such drift is detected, the DT may need to be updated, recalibrated, or even redesigned to maintain its validity. This introduces the concept of *temporal correctness* (representational fidelity over time, not real-time timing). A DT may be correct at one point in time but become incorrect as the original system evolves. Continuous validation mechanisms, including anomaly detection, residual monitoring, and periodic recalibration, are necessary to maintain alignment between the DT and its original system, as they ensure that the DT remains representative of the evolving original system.

Verification must consider how changes in the original system, the sensors, data pipelines etc. affect the accuracy of the DT's representation. Sensors degrade, data pipelines may be disrupted and temporarily lose data, and the original system may evolve in ways that are not captured by the DT's models. Verification mechanisms such as runtime monitoring, consistency checking, and integrity verification of data pipelines are required to ensure that the DT continues to operate correctly at despite ongoing changes.

Compositional Dimension *Validation* must particularly consider federated DTs and their ontologies for representing their original systems. Without alignment, the resulting federation may behave inconsistently, even if each individual DT is valid on its own.

Verification must consider two aspects: (i) *federation* of DTs requires verification that the resulting overall DT correctly implements a DT of the real system of systems, for example, where data is ingested at different granularity or fidelity; and (ii) *instantiation* of DTs from digital models (e.g., a model of a particular make and model of car being instantiated for each specific car of this type) requires verification of the instantiation process to ensure it correctly binds model components to real-world entities and calibrates parameters using instance-specific data.

4 Research Challenges

Building on the two lenses and three dimensions, we identify open research challenges for the V&V of DTs, focusing, particularly, on *how* to verify and validate the properties described in Section 3.

Lens 1 Challenges

Functional Dimension

(C1) Integration of V&V efforts. How to undertake holistic V&V across code, data pipelines, learning systems to ensure overall fitness-for-purpose and correctness? How can we test behaviour that was not explicitly designed?

(C2) Validity in context of use. How to define, justify, and update validity claims that are specific to decision type, operating regime, time horizon, and target population, rather than treating validity as a single global property?

(C3) Human-in-the-loop effects. Which V&V artifacts are needed by different stakeholders (e.g., operators, maintainers, and regulators), and how can they be kept mutually consistent?

Temporal Dimension

(C4) Dynamic assurance. How can V&V obligations be selectively re-triggered when models, data, requirements, or stakeholders change?

(C5) Governance, traceability, and evidence management. How can assurance maintain a traceable chain from assumptions to code version, datasets, intended use, and approval status over time [5, 14]?

Compositional Dimension

(C6) V&V across federations. How can verification enforce correct DT behaviour across federated twins? How can validation test federation-level fitness-for-purpose and prevent purpose creep when new twins are added?

Lens 2 Challenges

Functional Dimension

(C7) Uncertainty and confidence. How should validation quantify and communicate uncertainty from sensing, model structure, epistemology, and human inputs, and map it to risk-based acceptance criteria? How can uncertainty quantification and propagation be reliably verified?

(C8) Identifiability and observability. How can validation distinguish mechanistic credibility from mere data fit when latent states are unobservablee [18]?

(C9) Transfer problem. In the absence of ground-truth oracles [2, 24], how can validation ensure that models trained on observational data remain valid when deployed in operational contexts with potentially different data distributions?

Temporal Dimension

(C10) V&V when DTs alter behavior. How can validation justify intervention-stable counterfactual claims for "what-if" analysis [19]? How can validation separate twin adequacy from outcomes driven by altered operational behavior? When intervention changes trajectories, how can verification assess prediction quality?

Compositional Dimension

(C11) Cross-model representational coherence. How can validation maintain coherent claims as representations are composed across scales and subsystems, and how can verification preserve consistency of supporting evidence artifacts?

(C12) Semantic alignment. How can validation establish that coupled twins refer to the same entities and mechanisms, and how can verification check schema and mapping consistency?

Summary. Overall, DTs require a "living assurance framework" [4, 11] that links each validation claim and verification obligation to context of use, temporal evolution triggers, and compositional boundaries, supporting continual adaptation and re-verification / validation. Our conceptual framework can underpin assurance strategies for DTs by identifying the key properties to be verified and validated. The framework is also convenient for identifying and categorizing open research challenges for V&V of DTs.

Acknowledgments

This work was partly supported by the French government through the French National Research Agency (ANR) as part of the France

2030 program EDT (grant #ANR-25-APJN-0001). It also received funding from the KCL Informatics Support Fund. BoFan Zhang was partially supported by The China Scholarship Council (CSC, grant number 202408890021); Aidan Dures was partially supported by the Centre for Doctoral Training in Digital Twins for Healthcare with PhD funding by Synoptix Ltd. Christopher Burr was partially supported by UK Digital Twin Centre (Digital Catapult) Digital Twin Academic Accelerator grant DARTER (Digital twin assurance via runtime trust and evidence reporting).

References

[1] Bert Van Acker, Paul De Meulenaere, Hans Vangheluwe, and Joachim Denil. 2024. Validity frame-enabled model-based engineering processes. *Simulation* 100, 2 (2024), 185–226.

[2] Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. 2015. The Oracle Problem in Software Testing: A Survey. *IEEE Transactions on Software Engineering* 41, 5 (May 2015), 507–525.

[3] Julia Bitencourt, Ana Wooley, and Gregory Harris. 2024. Verification and validation of digital twins: a systematic literature review for manufacturing applications. *International Journal of Production Research* 63, 1 (2024), 342–370.

[4] Christopher Burr, Justin Anderson, Sophie Arana, Daniel Block, James Byrne, Ryan Goodman, Carlos Gavidia-Calderon, Ibrahim Habli, Nury Moreira, Steven Niederer, Nuala Polo, and David Wagg. 2024. *Trustworthy and Ethical Assurance of Digital Twins: Putting the Gemini Principles into Practice.* Technical Report. Alan Turing Institute.

[5] Christopher Burr and David Leslie. 2022. Ethical assurance: a practical approach to the responsible design, development, and deployment of data-driven technologies. *AI and Ethics* 3, 1 (June 2022), 73–98.

[6] Betty H. C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, Jeff Magee, Jesper Andersson, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Giovanna Di Marzo Serugendo, Schahram Dustdar, Anthony Finkelstein, Cristina Gacek, Kurt Geihs, Vincenzo Grassi, Gabor Karsai, Holger M. Kienle, Jeff Kramer, Marin Litoiu, Sam Malek, Raffaela Mirandola, Hausi A. Müller, Sooyong Park, Mary Shaw, Matthias Tichy, Massimo Tivoli, Danny Weyns, and Jon Whittle. 2009. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In *Software Engineering for Self-Adaptive Systems.* Springer B H, 1–26.

[7] Committee on Foundational Research Gaps and Future Directions for Digital Twins, Board on Mathematical Sciences and Analytics, Committee on Applied and Theoretical Statistics, Computer Science and Telecommunications Board, Board on Life Sciences, Board on Atmospheric Sciences and Climate, Division on Engineering and Physical Sciences, Division on Earth and Life Studies, National Academy of Engineering, and National Academies of Sciences, Engineering, and Medicine. 2024. *Foundational Research Gaps and Future Directions for Digital Twins.* National Academies Press, Washington, D.C. Pages: 26894.

[8] Rogério de Lemos, Holger Giese, Hausi A. Müller, Mary Shaw, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Gabriel Tamura, Norha M. Villegas, Thomas Vogel, Danny Weyns, Luciano Baresi, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Ron Desmarais, Schahram Dustdar, Gregor Engels, Kurt Geihs, Karl M. Göschka, Alessandra Gorla, Vincenzo Grassi, Paola Inverardi, Gabor Karsai, Jeff Kramer, Antónia Lopes, Jeff Magee, Sam Malek, Serge Mankovskii, Raffaela Mirandola, John Mylopoulos, Oscar Nierstrasz, Mauro Pezzè, Christian Prehofer, Wilhelm Schäfer, Rick Schlichting, Dennis B. Smith, João Pedro Sousa, Ladan Tahvildari, Kenny Wong, and Jochen Wuttke. 2013. Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. In *Software Engineering for Self-Adaptive Systems II.* Springer Berlin Heidelberg, 1–32.

[9] Romina Eramo, Francis Bordeleau, Benoit Combemale, Mark van den Brand, Manuel Wimmer, and Andreas Wortmann. 2022. Conceptualizing Digital Twins. *IEEE Software* 39, 2 (2022), 39–46.

[10] Michael Grieves. 2023. Digital Twins: Past, Present, and Future. In *The Digital Twin*, Noel Crespi, Adam T. Drobot, and Roberto Minerva (Eds.). Springer International Publishing, Cham, 97–121.

[11] Richard Hawkins, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu, and Ibrahim Habli. 2021. *Guidance on the Assurance of Machine Learning in Autonomous Systems.* Technical Report. Assuring Autonomy International Programme (AAIP), University of York.

[12] Edward Y Hua, Sanja Lazarova-Molnar, and Deena P Francis. 2022. Validation of Digital Twins: Challenges and Opportunities. In *2022 Winter Simulation Conference (WSC).* IEEE, 2900–2911.

[13] International Organization for Standardization 2005. *Quality management systems — Fundamentals and vocabulary.* International Organization for Standardization, Geneva, Switzerland. https://iso.org

[14] Timothy Patrick Kelly. 1998. *Arguing Safety – A Systematic Approach to Managing Safety Cases.* Ph. D. Dissertation. University of York, Department of Computer Science. https://scsc.uk/documents/acwg/tpk/Arguing%20Safety.pdf

[15] J.O. Kephart and D.M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50.

[16] Hussein Marah and Moharram Challenger. 2025. (Re-)Engineering Digital Twins Towards Federation: Vision and Roadmap. In *Leveraging Applications of Formal Methods, Verification and Validation. Software Engineering Methodologies*, Tiziana Margaria and Bernhard Steffen (Eds.). Vol. 15222. Springer Nature Switzerland, Cham, 60–81.

[17] Joost Mertens and Joachim Denil. 2025. Reusing Model Validation Methods for the Continuous Validation of Digital Twins of Cyber-Physical Systems. *Software and Systems Modeling* 24, 5 (2025), 1427–1449.

[18] Steven A. Niederer, Michael S. Sacks, Mark Girolami, and Karen Willcox. 2021. Scaling digital twins from the artisanal to the industrial. *Nature Computational Science* 1, 5 (2021), 313–320.

[19] Judea Pearl. 2009. Causal Inference in Statistics: An Overview. *Statistics Surveys* 3, none (Jan. 2009).

[20] Ana Perisic and Branko Perisic. 2024. Digital Twins Verification and Validation Approach through the Quintuple Helix Conceptual Framework. *Electronics* 13, 16 (Aug. 2024), 3303.

[21] M. Raissi, P. Perdikaris, and G.E. Karniadakis. 2019. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* 378 (Feb. 2019), 686–707.

[22] Kaan Sel, Andrea Hawkins-Daarud, Anirban Chaudhuri, Deen Osman, Ahmad Bahai, David Paydarfar, Karen Willcox, Caroline Chung, and Roozbeh Jafari. 2025. Survey and perspective on verification, validation, and uncertainty quantification of digital twins for precision medicine. *npj Digital Medicine* 8, 1 (Jan. 2025), 40.

[23] Concetta Semeraro, Mario Lezoche, Hervé Panetto, and Michele Dassisti. 2021. Digital twin paradigm: A systematic literature review. *Computers in Industry* 130 (2021), 103469.

[24] E. J. Weyuker. 1982. On Testing Non-Testable Programs. *Comput. J.* 25, 4 (Nov. 1982), 465–470.