Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

THE PRESENT AND EVOLUTION OF TWINNING: RETHINKING THE MULTIFACETED REPRESENTATIONS OF COMPLEX SYSTEMS

Andrea D'Ambrogio[1], Edward Y. Hua[2], Giovanni Lugaresi[3], Guodong Shao[4], Mamadou K. Traoré[5], and Hans Vangheluwe[6,7]

[1] Dept. of Enterprise Eng., University of Rome Tor Vergata, Rome, ITALY
[2] The MITRE Corporation, McLean, USA
[3] Dept. of Mechanical Eng., KU Leuven, Leuven, BELGIUM
[4] Eng. Laboratory, National Institute of Standards and Technology, Gaithersburg, USA
[5] Dept. of Engineering and Digital Sciences, University of Bordeaux, Talence, FRANCE
[6] Dept. of Computer Science, University of Antwerp, Antwerp, BELGIUM
[7]Flanders Make at the University of Antwerp, Antwerp, BELGIUM

ABSTRACT

The concept of *twinning* has gained significant traction across industry and academia, largely driven by the rise of digital twins (DTs). DTs have become ubiquitous in fields such as manufacturing, healthcare, and urban planning, with applications tailored to specific domains—from physical building replicas for design and fundraising to digital factory models for real-time operations. While twinning research has focused primarily on the "digital" aspects of a DT through architectures, frameworks, and technical implementations, this panel examines the "twin" part of a DT, emphasizing the modeling process from a system to its representation, which is at the heart of *twinning*. The discussion focuses on fidelity levels, validity frames, and terminological nuances. With examples, the panel emphasizes the need for formal modeling foundations, scalable workflows, and interoperability to support twinning systems. The aim is to explore future research directions and novel applications of twinning.

1 INTRODUCTION

Twinning has recently emerged as a transformative concept in both academic research and industrial practice, bridging the physical and digital realms to enable deeper understanding, control, and optimization of complex systems. The notion of creating mirrored representations of real-world entities is not new (e.g., physical scale models or mathematical simulations). However, the recent surge in data availability and computational power has elevated twinning to a new level of dynamism and precision. Digital Twins (DTs) are the most prominent instantiation of this paradigm, since they are based on a real-time synchronization between systems under study and their models, fostering applications that span from predictive maintenance in manufacturing to proactive urban planning and adaptive business process management. DTs are used in a variety of domains (Dalibor et al. 2022). They address many industrial needs, such as anomaly detection, condition monitoring, predictive maintenance, and performance optimization. Yet, despite their omnipresence, many different and sometimes contradicting definitions of DT exist (Semeraro et al. 2021).

As twinning becomes increasingly central to cyber-physical system design and operation, there is a growing need to revisit its conceptual foundations, refine its terminology, and explore its evolving dimensions across domains. The *twinning paradigm* spans the entire spectrum between Modelling and Simulation and Internet of Things thus subsuming Digital Models, Digital Shadows, Digital Twins, Physical Models, Physical Twins, *etc.*

This panel brings together diverse perspectives to explore the theoretical, methodological, and practical dimensions of twinning, with a focus on fidelity, validity, and interoperability across domains. To foster a deeper understanding and stimulate cross-disciplinary dialogue, a group of experts reflects on a curated set of questions that probe the challenges and opportunities of twinning in both research and applied contexts. The panel organizers identified the following questions:

1. Summarize a use case or application of twinning you have experience with.
2. How to combine multiple twinning architectures?
3. How are twinning and Modelling related?
4. How to deal with the limited validity range of models in twinning?
5. How to distinguish between a system anomaly and the model in the twin object going out of validity range?
6. How to deal with evolution (e.g., of twinning goals, of System under Study models, of deployment technology)?
7. How to deal with hard real-time constraints, and the need for guarantees?

2 THE COMMON DENOMINATOR: TWINNING SYSTEMS

A twinning system connects a System under Study (SuS) with a model of that SuS in some appropriate formalism (and its simulator/executor) (Grieves 2014) and keeps them *continually* synchronized. In such a twinning system, we call the Actual Object (AO) a conceptual, abstracted view on and interface to the SuS and its environment. Similarly, a component based on the representation of the SuS and its environment is called a Twin Object (TO). To structure the creation of twinning systems, a high-level workflow is proposed, shown in Figure 1. It is based on the IIRA Architecture Framework (Industry IoT Consortium 2022) and the ISO/IEC 12207 standard (Singh 1996). This workflow is comprised of four inter-dependent stages, listed in the following.

Stage A: Making Choices in the Problem Space. This first stage represents the identification of requirements. It is meant to answer *why* a twinning system must be constructed. One may classify these reasons or requirements. Kang and Lee (2013) distinguishes between *goals*, *usage contexts* and *quality assurance*. There are many possible reasons for constructing a twinning system as found in the literature (Dalibor et al. 2022; Van der Valk et al. 2020; Jones et al. 2020; Wanasinghe et al. 2020; Minerva et al. 2020). Figure 2 shows a non-exhaustive collection, known as a feature model, of possible goals. Connections with filled/open circles identify mandatory/optional features. The full arrows identify dependencies and the dashed arrows denote optional dependencies. Every path through the feature tree, corresponding to a sequence of choices made during requirements analysis of a twinning system, leads to a goal. For example, Operation $\rightarrow$ Data Processing/Analysis $\rightarrow$ Anomaly Detection. Once this goal has been selected, The Properties of Interest (PoI) over which in this case Anomaly Detection is required still need to be selected. For example, in a vehicle a PoI may be velocity: when actual velocity deviates too much from expected velocity (i.e., as predicted by a TO), an anomaly is reported. A more elaborate breakdown of goals, usage contexts and quality assurance can be found in Paredis and Vangheluwe (2024).

Stage B: (Conceptual) Architecture and Design. Once choices have been made in stage A, individual system components required to (functionally) ensure the valid behaviour of the twinning system need to be identified and combined, at a conceptual level. Stage B answers to *what* is required for a twin to be constructed. The diagram at the bottom of stage B in Figure 1 shows a conceptual reference architecture, following the IIRA's *functional viewpoint*. The figure is annotated with numbered variation points, also known as presence conditions, to include or remove components and/or connections, based on the choices made in stage A.

The *Actual Object* (AO) 1 is an *abstraction* of a specific *view* of the actual world (and environment) in which the AO is active. *Twin Object* (TO) 2 is a component that is continually kept synchronized with the SuS, with respect to the chosen goals and PoIs. For example, the TO can be a neural network, the execution of code, or a (real-time)simulator for a model in an appropriate formalism. The formalism(s) and model(s) of the TO are to be chosen in stage C. The Experiment Manager (EM) 3 contains a *workflow* 4, a model of *how* the experiment is to be executed. An experiment is an intentional set of (possibly hierarchically composed) activities, carried out on a specific SuS in order to accomplish a specific set of goals. Each experiment should have a description, setup and workflow, such that it is (at least) *repeatable* (Plesser 2018). Since the experiment is created for a specific set of requirements, the requirement's logic is contained in 3. For instance, if we only want to have a dashboard to visualize the current state, the collection of this state is done by the EM. If instead our goal is Anomaly Detection, the EM needs to compute the distance between the behaviour observed by the Actual Object and the behaviour computed by the Twin Object – typically over a moving time window – and produce a notification when this distance exceeds a given threshold. 5 and 6 denote the communication between the EM and the AO or TO, respectively. The Experiment Manager can communicate with a User Agent (7A) or Machine Agent (7B), access points for a user or another system to obtain information about/from or send information to this experiment.

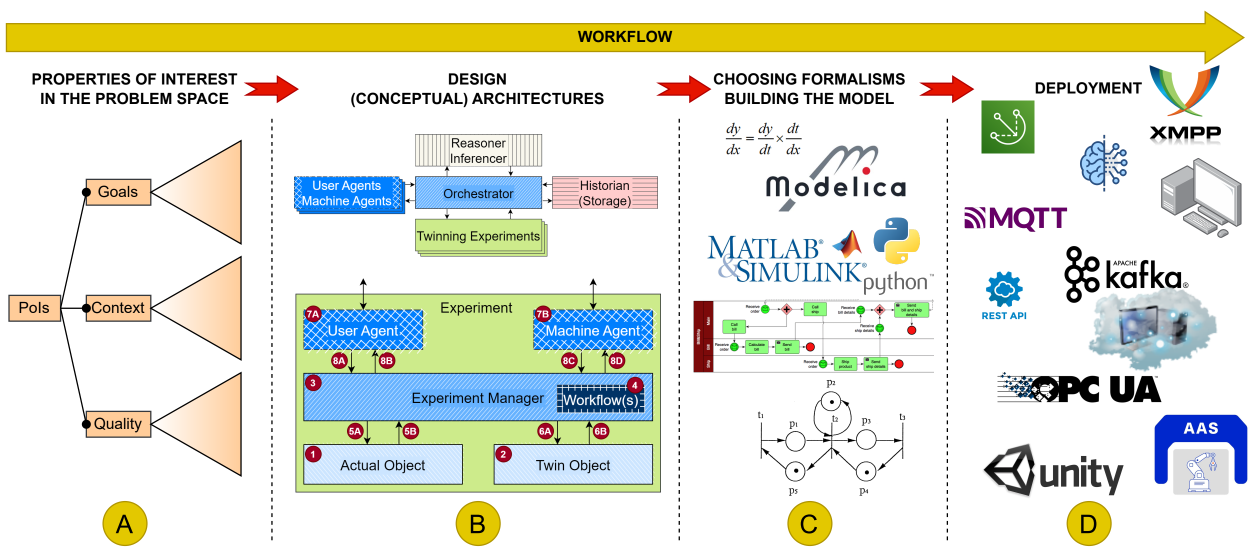

Figure 1: Generic workflow for creating twinning architectures.

Stage C: Formalisms and Models. In this phase, most appropriate modelling languages are chosen in models pertaining to the goals and PoIs of the SuS selected in stage A are built. How to do this is the expertise of the modelling and simulation community.

Stage D: Deployment. Finally, the conceptual architecture from stage B needs to be realized. For this, frameworks, languages, middleware, communication protocols etc. need to be selected. How to do this is the expertise of the systems engineering and deployment community. The particular choices made here lead to the (software) architectures often seen in the large DT literature.

3 PERSPECTIVES FROM ANDREA D'AMBROGIO: TWINNING FOR BUSINESS PROCESS MANAGEMENT

Twinning is increasingly being applied to *business process management* (BPM), in addition to more conventional applications to manufacturing and (cyber-)physical systems. Business process models and their execution over time (i.e., simulation) are traditionally being used as offline tools applied to various stages of the business process lifecycle, from process definition and design, to visualize, analyze and evaluate the impact of design choices, down to the process execution and monitoring, to optimize workloads and deal with unexpected events or performance downgrades. Recently, the availability of large amounts of data collected by so-called *process-aware information systems*, which provide features to record data about process executions, as well as their archiving in event logs, has paved the way to *process mining*, a set of techniques that have been introduced in the BPM domain for analysing event data to understand and improve the performance of operational processes. The relevance of process mining for the construction and use of twinning systems is emphasized in (Fornari et al. 2024).

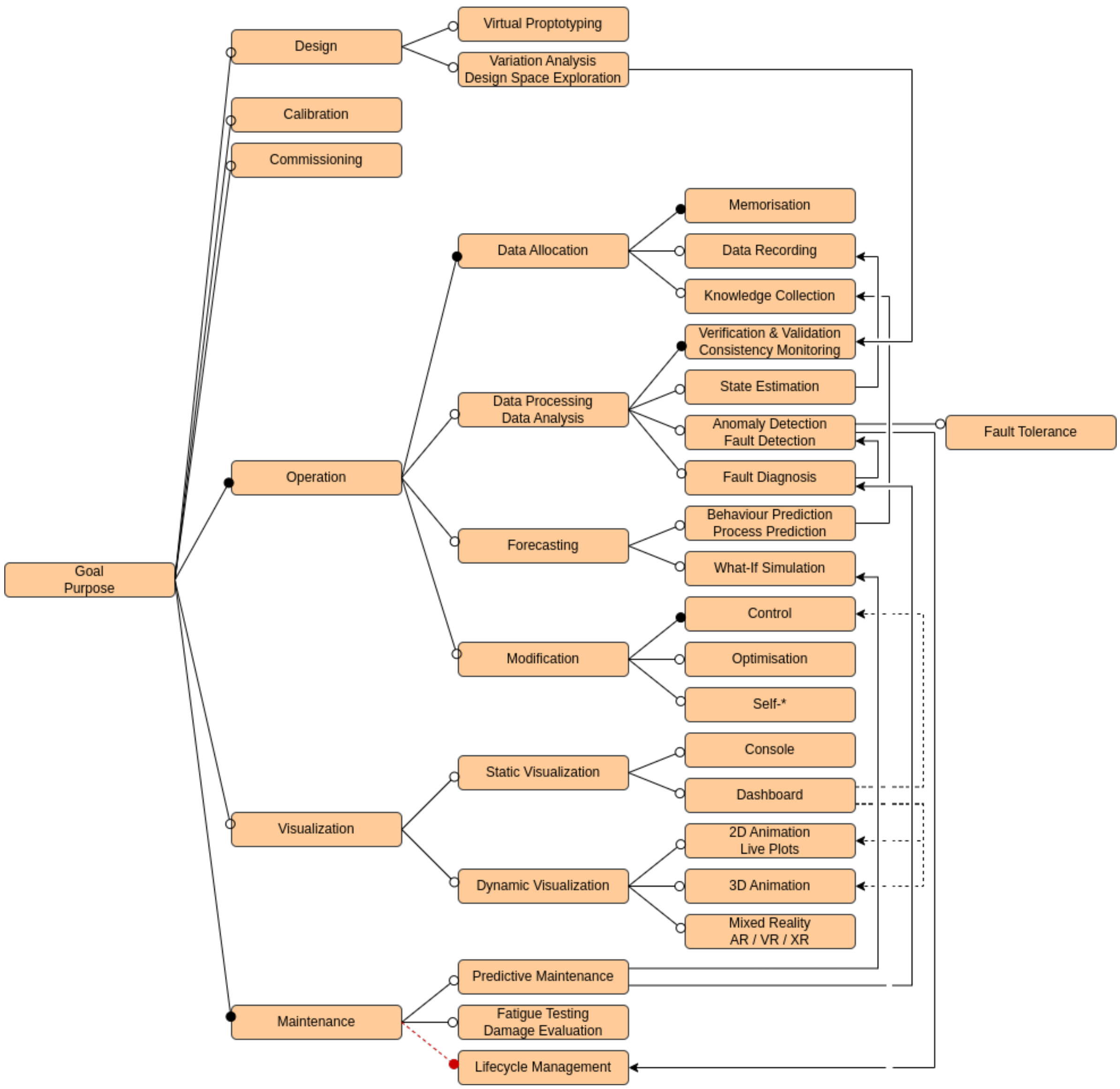

Figure 2: A feature model describing possible goals of the twinning system.

The combined use of process mining techniques and tools, which focus on past process executions, thus enabling a retrospective process analysis, and *modelling and simulation* approaches, which provide capabilities to carry out predictive process analysis based on what-if scenarios, can be exploited to create business process DTs based on real-time, data-rich models that mirror the actual workflows, resource allocations, and performance metrics. These twins can be used as online tools for continuous monitoring, predictive analytics, and rapid experimentation without disrupting actual process operations, with the goals of virtual prototyping (Design in Figure 2), identifying inefficiencies (Operation in Figure 2), test improvements (Maintenance in Figure 2), and/or respond proactively to changing conditions (Operation in Figure 2).

The typical PoIs for process twinning include *structural properties*, such as the roles and resources involved, *behavioral properties*, such as task durations, waiting times, process cycle times, and *performance metrics*, such as throughput, task completion rates, resource utilization, resource reliability and costs. Figure 3 introduced and illustrated in (Bocciarelli and D'Ambrogio 2024), proposes the architecture of a *simulation-based predictive process mining* approach that is at the core of a business process twinning system, along with the formalisms and models used, as well as the relevant deployment technologies. As an example, the approach could be applied to business processes for predictive maintenance. Through the integration of IoT sensors, process mining algorithms and simulation-based techniques, according to the steps shown in Figure 3, the twinning system is able to forecast potential failures and recommend optimal maintenance actions, so to improve the overall reliability, reduce operational costs, and extend the useful life of critical assets. The twin object enables proactive decision making in case of predicted resource failures, by providing preventive maintenance notifications to be dealt with manually by ensuring minimal impact on system operations and by triggering the corresponding automated reconfiguration of the twin object.

Question n.2 (*How to combine multiple twinning architectures?*) deals with a significant challenge for twinning applied to business process management. Modern business processes operate within a complex network of interconnected subsystems, each contributing to the overall operational efficiency. Organizations engaging in collaborative tasks for collectively contributing to an overall business process define so-called *process collaborations*, which are based on the interchange of information and data among distributed participants. The collaboration provides each participant the opportunity to create value by interoperating with other participants without however disclosing the details of their internal business processes. In this scenario, the ability to federate possibly heterogeneous twinning architectures is essential to properly manage the potential of opening to collaboration for organizations that plan to share their resources in collaborative workflows. In this respect the research work carried out recently to evaluate the potential of distributed simulation approaches (based, e.g., on the High Level Architecture standard) for analyzing process collaborations can be seen as a starting point to define *federated architectures for collaborative twinning* (El Kassis et al. 2024).

Question n.3 (*How are twinning and modeling related?*) is highly relevant for business process twinning. Modeling is at the core of a twin object, not only to effectively deal with the various levels of abstraction and multiple languages applied to build and execute a business process twin, but also to exploit the significant degree of automation implied by formal model-based approaches. The use of model-to-model and model-to-text transformations, code generation and model interpretation provides significant opportunities to reduce the effort and improve the quality at twin design, development and operation stages, but also poses important challenges (Lehner et al. 2025). In the business process management domain, process definition is facilitated by the availability of a standard and widely used modelling language (i.e., the BPMN), which however has to be properly interpreted, extended and customized to be part of twinning environments consisting of heterogeneous data, models and tools (e.g., workflow orchestration engines, event logs, IoT devices, model parameters, process mining tools, simulation engines, etc.), as partially shown in Figure 3.

Question n.4 (*How to deal with the limited validity range of models in twinning?*) and question n.5 (*How to distinguish between a system anomaly and the model in the twin object going out of validity range?*) are directly linked to the previous question. Ideally, the inherent discrete-event nature of a business process and the availability of event logs should make easier spotting whether the model in the twin object is no more valid or an actual object anomaly requires a twin intervention. However, the heterogeneity of data formats, timestamps, models and protocols, as well as temporal delays, may make it harder to achieve an accurate synchronization of the twin object.

Question n.6 (*How to deal with evolution of twinning goals, of System under Study models, of deployment technology, etc.?*) can be framed in the general context of software and systems engineering. Building and

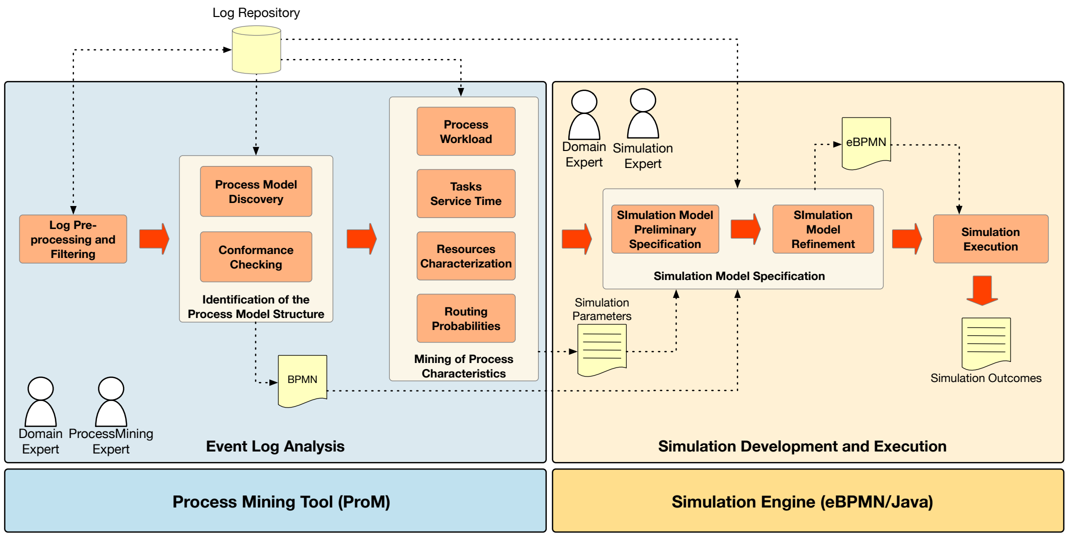

Figure 3: Simulation-based predictive process mining for business process twinning (from Bocciarelli and D'Ambrogio 2024).

maintaining a twinning system is by itself a complex engineering effort, beyond the specific application domain. It is thus essential being able to differentiate between technical complexity, which can be dealt with innovative data-driven and model-driven solutions, and essential complexity, which needs further research to effectively address scalability and evolvability issues (Fornari et al. 2024).

Finally, question n.7 (*How to deal with hard real-time constraints?*) is in my opinion relevant for a subset of business process twinning systems, specifically the ones interfacing with IoT devices that are used to monitor and drive process execution in real-time. In such a case, the acquisition of data from distributed and heterogeneous IoT devices requires events processing approaches able to efficiently deal with the different granularity and format of the various event streams that need to be synchronized.

4 PERSPECTIVES FROM GUODONG SHAO: THE ROLE OF STANDARDS

The validity range of a digital twin refers to the specific conditions under which its models accurately represent the physical system. It includes factors such as input ranges, operating environments, and assumptions. As physical systems change over time due to wear, upgrades, or new configurations, models can become outdated. Using a digital twin/model outside its intended scope can lead to incorrect predictions, safety risks, and loss of trust. Defining clear boundaries of applicability, monitoring, and managing drift helps prevent misuse and supports responsible decision-making.

The ISO standard, *ISO 23247 Digital Twin Framework for Manufacturing*, defines a digital twin in manufacturing as a "fit for purpose digital representation of an observable manufacturing element (OME) with synchronization between the element and its digital representation"(ISO 2021a). The OME could also be referred to as a System under Study (SuS). Fit-for-purpose requires that each digital twin has its own scope and objective, context of use, and relevant data and models. We need to clearly define the model's operating envelope, implement continuous monitoring for changes in input data or system behavior, and trigger retraining or recalibration as needed. The synchronization between the model and its physical system keeps the digital twin connected, but it alone cannot sufficiently guarantee model validity. Ideally, an intelligent digital twin should be able to learn and adjust automatically based on the data updated through synchronization and rules. However, techniques for setting such rules, monitoring, and incorporating human-in-the-loop are still needed to ensure that the digital twin remains valid and robust. Standards and frameworks can help organizations manage the limited validity range of digital twins by providing structured guidance, common terminology, verification and validation procedures, and best practices across industries. By embedding validity management throughout its lifecycle, digital twins can stay reliable, trustworthy, adaptive, and transparent even as real-world conditions change.

4.1 How to Deal with the Limited Validity Range of Models in Twinning?

Models work best within a known operating envelope and conditions on which they were trained, and a temporal window where system behavior remains stable. Maintaining validity is not a one-time task; it requires attention across all phases of the digital twin lifecycle: from requirements to conceptual design, formalism selection, deployment, and operation (Akcay et al. 2023). Verification, Validation, and Uncertainty Quantification (VVUQ) standards and techniques can help identify issues and manage the limited validity range of digital twins. Verification confirms the model equations, code, and data pipelines behave as intended within the declared validity envelope. Validation tests the model against real-world data only inside its accepted operating conditions and flags when the system or environment moves outside that range. Uncertainty Quantification (UQ) assigns confidence bounds or probability distributions to predictions; those bounds expand (or confidence scores drop) when inputs drift beyond the model's valid domain, signaling the need for recalibration or retraining. VVUQ is a formal process to declare, test, monitor, and communicate where a model is trustworthy. All activities aimed at defining the envelope, detecting drift, triggering retraining, and providing fallback logic are considered part of VVUQ. Without this validity-range management, a digital twin cannot be said to be fully verified, validated, or uncertainty-quantified. Figure 1 shows the VVUQ interactions with the digital twin lifecycle, and activities in each phase are briefly discussed as follows:

- Stakeholders' requirements phase: Defines fit-for-purpose validity expectations for a specific use case, clarifies what the digital twin is expected to do and under what conditions it must remain valid, specifies the operational envelope, quantifies the input and output ranges, and identifies critical uncertainties and risks that may lead to serious consequences due to out-of-validity predictions.
- Development phase: Designs for modularity and flexibility, builds model architectures that allow for extensions as the system evolves, includes fallback mechanisms such as plans for what the digital twin should do when it's operating outside the known validity range, and plans for detecting drift or out-of-validity conditions, quantifies the validity envelope, uses domain knowledge, statistics, and simulation to define input/output boundaries, represents uncertainty formally, and enables traceability and explainability. A hybrid modeling approach may be used so that different models can handle distinct operating regimes (e.g., high-load vs. low-load) in real time.
- Deployment phase: Ensures the digital twin works as expected in the real-world environment. The acceptance verification confirms that the right model version, parameters, units, and data interfaces are what were tested in pre-production. The commissioning validation compares predictions and ground truth, checks performance against acceptance criteria (e.g., $\leq 5\%$ error, $\leq 1s$ latency). The baseline UQ and validity envelope capture reference error statistics, input distributions, and confidence bounds that represent the "healthy" state on Day 1, and log system configuration, sensor firmware, and process settings to anchor future comparisons.
- Maintenance and operation phase: Ensures the digital twin is trustworthy over time, catches degradation early, and guarantees that every post-change model is re-verified and re-validated before it influences decisions. Continuous performance monitoring tracks prediction error, latency, and availability. Data and concept drift detection monitors input distributions and flags unseen operating regimes or out-of-envelope inputs. UQ in real time attaches confidence scores to every prediction, widening intervals as the model ventures toward boundary conditions. When firmware, process logic, or sensors change, automatically rerun targeted verification tests. If drift persists, retraining or parameter tuning will be triggered, the validity envelope and baseline statistics will be redefined, and the digital twin version will also be updated. Traceability logs need to be generated

by recording who and what changed the model, when, and why, retaining evidence for regulators and root-cause analysis.

- Decommissioning phase: At the time of decommissioning, any updated validity envelope and changed conditions need to be documented and archived for potential future reuse of the digital twin models.

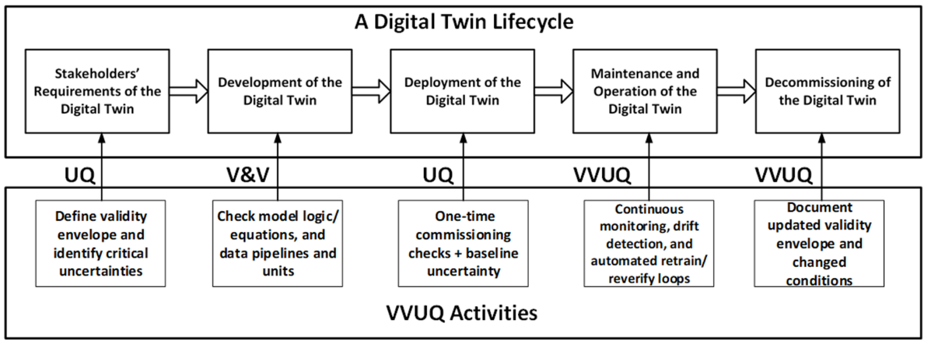

Figure 4: Verification, validation, and uncertainty quantification (VVUQ) in the DT life cycle.

4.2 How Can Standards Help Manage the Limited Validity Range of Models in Twinning?

During a digital twin operation, distinguishing between a physical system anomaly and the digital twin model drifting requires strategies that blend observability, diagnostics, and validation techniques. These strategies and techniques include (1) continuously comparing model predictions to real-world outcomes and monitoring for data distribution changes, (2) triggering retraining or recalibration by setting automatic or expert-verified rules to update the model when validity is compromised, (3) using hybrid models, i.e., combining data-driven models with physics-based ones to extend validity and increase robustness, and (4) maintaining model versions by aligning model versions with system configurations that includes validity ranges. Standards can help manage these processes by providing structured methods for validation, monitoring, and traceability throughout the digital twin lifecycle.

- Validity envelope specification: Standards, such as ISO 23247 (ISO 2021a; ISO 2021c) or IEEE P2806 (IEEE 2019) , can provide guidelines for specifying clear fit-for-purpose scope and objectives, data requirements, assumptions, and conditions for the digital twin. These are the foundations of the validity envelope. For example, if incoming data falls outside these documented bounds, it's likely the model is out of date or misaligned. Conversely, if the data is within the envelope but still shows a mismatch, it's likely a real anomaly in the physical system.
- Structured validation and verification (V&V) frameworks: Standards can support model VVUQ. The ASME V&V standards and ISO/IEC TR 24029-1 (ISO 2021b) define systematic approaches for assessing model accuracy, uncertainty, and performance across operating conditions. They can help establish confidence thresholds and error margins, so when a model prediction deviates from observed data, it can be compared against known validity limits to determine whether it's the model or the system at fault.
- Traceability and change logging: Standards such as ISO 15926 (ISO 2003) and ISO/IEC 25024 (ISO 2015) support detailed traceability of system changes, configurations, and model updates. If a discrepancy arises, and the system has undergone recent changes that are not reflected in the digital twin, it is likely that the model is outdated rather than the system failing. In addition, adaptive models can incrementally update themselves with new data. ISO/IEC 23894 (ISO 2023) provides guidance on risk management and helps improve robustness.

- Monitoring and drift detection guidance: Standards that address AI and data governance, e.g., ISO/IEC 22989 (ISO 2022) and ISO/IEC 24028 (ISO 2020), recommend the use of data drift detection and continuous monitoring. These techniques can show whether the input data distribution has shifted over time or whether the system behavior has changed unexpectedly.
- Feedback loops: By involving human-in-the-loop validation, experts can review alerts and compare behavior with operational knowledge. This supports interpretability and helps domain experts judge whether a deviation is explainable or signals a true fault or an outdated model. ISO 23247 specifies a user entity that allows human experts to enter their judgment (ISO 2021c), ISO/IEC 24029-1 describes ways to evaluate AI model robustness and integrate expert review when models behave unexpectedly (ISO 2021b).

5 PERSPECTIVES FROM MAMADOU TRAORÉ: TWINNING FOR URBAN MANAGEMENT

Urban Digital Twins (UDTs) have emerged as promising tools to enhance urban resilience and sustainability. Long-standing urban issues, which encompass the expansion of public transport networks, energy grids, waste management circuits, and measures to combat the spread of disease and preserve air quality, are exacerbated by the acceleration of global urbanization, with nearly 70% of the global population expected to reside in urban areas by 2050 (UN-Habitat 2022).

5.1 Reference Use Case: Urban Digital Twins

An UDT is an integrated model-based and data-driven approach, which assimilates complex data from multiple sources into behavioural models and simulate "what-if" scenarios to optimize urban and periurban planning decisions.

Goal and properties of interest. In the following we refer to a multi-scale and multi-perspective UDT development project aiming at offering the public domain with UDT-based predictive and prescriptive services and immersive solutions. A multi-scale UDT is a UDT which models span from an individual level to building, district and city levels. A multi-perspective UDT is a UDT that integrates different phenomenon of the built, natural, and social environment, encompassing not only the mobility and transportation, but one or more other urban concerns, including energy use and production, water distribution, waste management, and urban asset maintenance.

Formalisms and models used. To formally specify a UDT, we adopt a system-theoretic approach (Zeigler et al. 2000; Traoré and Muzy 2006), where the structure of a model is defined as a black box with inputs and outputs interfacing the model with its environment (in the case of a UDT, the interface is realized with sensors and actuators). As shown at the left-hand side of Figure 1, basic entities are the source system (i.e., the system under study), the context, the model, and the experimental frame (EF) – a last entity, the simulator, is not represented here. The system under study acts as a source of behavioral data. The context is the set of conditions under which the system is being observed. The model is a set of rules or mathematical equations that give an abstract representation of the system, which is used to replicate its behavior. Equally, the EF, which gives an abstract representation of the context, is a model component to be coupled with the system model to produce the data of interest under specified conditions. The simulator is the automaton that is able to execute the instructions of the resulting coupled model. In contrast, a UDT model (right-hand side of Figure 3) once built is regularly updated with data collected from the source system. Hence, an inference mechanism is needed to allow such a regular update of the model, in order to preserve its validity within the same EF.

We then formally specify our UDT framework (also called the DMSμ framework) $SysUDT = \{D, M, S, \mu\}$, where: (i) D is the set of collected data; (ii) $M \in \mathbb{M}$ is the UDT model of the smart territory, which is specified as a DEVS model; (iii) $S \in \mathbb{M}$ is the UDT service, which is parametrized by M and D, and specified as a DEVS model; and (iv) $\mu : \Omega_D \times \mathbb{M} \rightarrow \mathbb{M}$ is the data assimilation function, where Ω_D is the set of admissible data segments and $\mathbb{M}$ is the set of all possible models. This formal approach

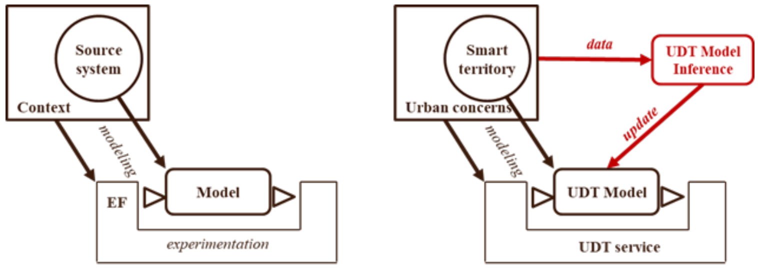

Figure 5: Traditional and urban digital twin.

enables the identification of different levels of self-updating capabilities, i.e., the ability of the μ function to detect and respond to changes, either in values (low level of knowledge) or in structure (high level of knowledge) of specific elements within the formal specification of M (Diakité and Traoré 2024), using statistical inference estimation techniques (such as Data fitting, Bayesian Inference, Kalman Filter, etc.).

Deployment technologies and architectures. The operational framework is a layered architecture, which starts from the smart territory's physical ecosystem by incorporating the processes, events, or activities within urban spaces. It then subsequently incorporates the Data, Models and Services core components of the UDT, as well as the human elements of stakeholders involvement, as shown in Figure 6:

- The Data layer is the communication gate for the Physical-Digital interactions (i.e., between the cyber physical system of the smart territory and the UDT). The Data layer relates to data from various sources, including real-time collection from sensors on ground, legacy data sources (e.g. smartphones, embedded cameras), open data and Internet APIs (e.g. Google map). It not only handle data, it also addresses how data can be assimilated for detecting changes in the real system and real-time updating (thus realizing the μ function).
- The Model layer is a gateway for Digital-Human interactions, enabling citizen engagement via avatars in the models (i.e., Human-in-the-loop participating to UDT models through a virtual/augmented reality device). The Models layer is a set of abstractions of reality, each designed to answer specific questions related to mobility, energy, etc. It includes descriptive, predictive, and prescriptive models. Descriptive models provide a detailed description of city phenomena as they exist or occurred in the past. GIS (Geographical Information System), BIM (Building Information Model), and CIM (City Information Model) are the existing mature descriptive data models of the urban ecosystem that provide information about the city phenomena and are important foundations for creating a 3D territory model/platform. However, the utilization of these models is not necessary for building a UDT. Predictive models make forecasts and predictions about future territory events and outcomes. Prescriptive models go beyond the prediction and offer the recommended actions or strategies to achieve the desired outcome in a territory.
- The Service layer is a Digital-Human interactions gateway that enables UDT commissioning by urban decision-makers (i.e., Human-out-of-the-loop sollicitating UDT services through a web-based/standalone interface). The Services layer leverages the UDT models the same way an EF leverages a model in the DEVS paradigm, for the provision of prediction, what-if scenarios exploration, diagnosis, and optimization services. It addresses how models are used/combined to provide the expected services, as well as, how services can be provided directly from data (e.g. monitoring of historical data).

5.2 How to Combine Multiple Twinning Architectures?

The DMSμ framework suggests a high-level symbolic representation for UDTs (and more generally for DTs), a molecular structure that is a symbiotic relation between data, models and services, which we use to

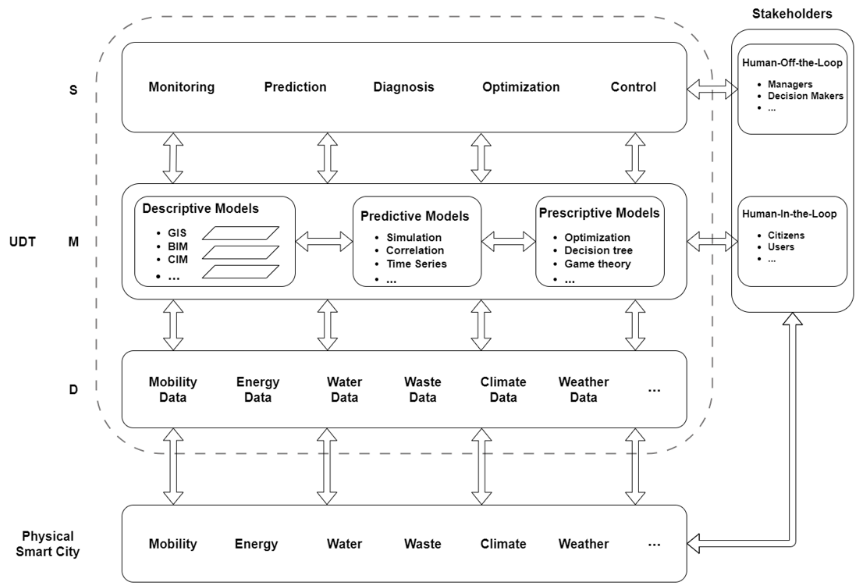

Figure 6: Multi-scale and multi-perspective UDT framework.

reason and symbolically manipulate UDT concepts, including modelling larger molecular constructions of UDTs as Systems of Systems (SoS). The strength of this symbolic molecular model is that it describes what has to be done and not how this has to be done, therefore giving flexibility in the choice of the technologies to be used and the way they will be. As several matured technologies already exist, they can be leveraged to serve this purpose. Figure 7 gives the various categories of such larger molecular constructions, which we call levels of UDT interoperability (Traoré 2023), each of which refers to UDT composition by an adapted mechanism of interoperability between two nodes of the model: (i) Data interoperability only involves Data nodes, thus dealing with data format conformance as well as semantic alignment; (ii) Model interoperability only involves Models nodes, thus dealing with multi-paradigm integration (i.e., multi-formalism, multiple temporal/spatial scales, multiple abstractions); an adequate way to address this is the hybridization strategies in computational frameworks introduced in (Tolk et al. 2018); (iii) Service interoperability only involves Services nodes, thus dealing with interoperability strategies such as service orchestration (where one of the services takes on the role of the orchestrator and coordinates the communication between all services involved) and service choreography (where services participate asynchronously and autonomously to a defined scenario); Standards exist (Barros et al. 2005) that can be leveraged to address this level of UDT interoperability; (iv) Data/Model reuse involves the Data node at one side and the Models node at the other side, thus addressing the questions of data reuse (i.e., the use of data for models that are not the ones for which the data were initially collected and consolidated) and model reuse (i.e., the use of a model with other datasets than the ones the model use to be fed with); in the case of data reuse, this level of interoperability cannot be achieved in the absence of metadata, which will provide a way to check not only the understandability of data, but also contextual information that refers to the set of interrelated environmental conditions in which data have been produced for the initial model; in the case of model reuse, a meta model is needed to provide the same kind of knowledge about the initial model; a potential way to address this level of interoperability is the experimental/validity frame approach (Traoré and Muzy 2006; Van Acker et al. 2019; Van Mierlo et al. 2020); (v) Data/Service reuse involves the Data node at one side and the Services node at the other side, thus addressing similarly the questions of data reuse and service reuse; (vi) Model/Service reuse involves the Models node at one side and the Services node at the other side, thus addressing similarly the questions of model reuse and service reuse.

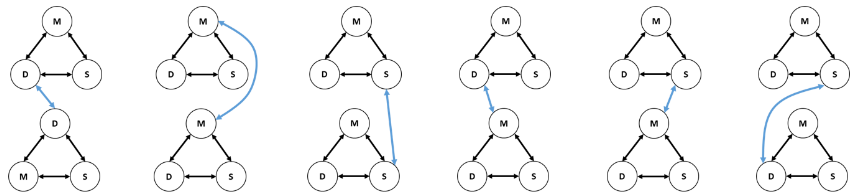

Figure 7: Levels of urban digital twins interoperability.

6 PERSPECTIVES FROM EDWARD HUA: THE IMPORTANCE OF VALIDATION

The twinning paradigm seeks to unify the tasks of designing, developing, and operating a DT into a cohesive and robust framework. Unlike traditional approaches that treat DT development as an isolated or siloed endeavour, twinning emphasizes a comprehensive view of multiple systems engineering practices, such as requirements engineering, experimentation, prototyping, design optimization, product deployment, and end-of-life management. This approach compels the DT engineer to consider the entire design, development, and operations stages of the DT, ensuring that the DT is robust and functional.

6.1 How are Twinning and Modelling Related?

Modelling plays a central role in the twinning paradigm, and it should be viewed as a continuous process rather than a discrete task confined to specific stages. The quality of modelling evolves progressively, transitioning from coarse approximations to refined representations as the project progresses through the workflow in Figure 1. This iterative refinement ensures that the DT accurately mirrors the physical entity and its operational state, enabling the DT to perform tasks such as "what-if" analysis and scenario exploration effectively. One of the significant advantages of the twinning paradigm is its support for creating and maintaining a robust digital thread. The digital thread represents the continuous flow of data that links various stages of a product's lifecycle. This data flow can be accessed and consumed by the DT, undergoing frequent revisions to refine the dataset. A well-developed digital thread enhances traceability across the product lifecycle and fosters improved collaboration and communication among stakeholders.

6.2 How to Perform Verification and Validation of a Twinning Paradigm Implementation?

Verification and Validation (V&V) of a DT model is inherently challenging due to the dynamic nature of the model itself (Lugaresi et al. 2023). Unlike static models, DT models are subject to continuous data flows that may alter their properties during operation. Consequently, V&V becomes a critical problem in DT design, development, and operation. In the twinning paradigm, V&V should start early, at the Problem Space stage where relevant information is gathered, such as goals, context, and quality attributes of the physical entity being modeled. This foundational information enables the DT engineer to develop an initial set of performance metrics that are of interest to the physical entity. At the Conceptual Design Architectures stage, verification should be performed to ensure that the DT model accurately reflects specifications of the physical entity. Once the DT is deployed, validation must be performed periodically to ensure its operational performance remains closely aligned with its physical counterpart.

Four types of validation techniques are identified that can be applied to DT models (Hua et al. 2022): manual/visual inspection, property testing, model-based testing, and machine learning (ML)-based validation. These techniques offer different approaches to address the complexities of validation, depending on the specific requirements and challenges of the DT model. A key component in implementing a robust validation mechanism within the twinning paradigm is data uncertainty quantification. Given the dynamic nature of DT models and the continuous flow of data, quantifying uncertainties in the data becomes essential for ensuring the reliability and accuracy of the model. By addressing data uncertainty quantification, the validation process can account for variations and inconsistencies that may arise during the model's operation, thereby enhancing the robustness of the DT. The twinning paradigm establishes a strong relationship between modelling and the iterative development of DTs, emphasizing continuous refinement and integration across the product lifecycle.

It should be noted that in this paper, the term "Validation" is expressed to carry different meanings among the authors. On the one hand, it is used in the mechanical context of Verification, Validation, and Uncertainty Quantification (VV&UQ); on the other hand, it is also defined in the more traditional sense of Verification, Validation, and Accreditation (VV&A). Furthermore, questions surrounding the ethics of V&V in simulation have been raised by (Tolk et al. 2021). The nuances of what it means to "validate" warrant further discussion as we look to the future.

7 CONCLUSION

This article has presented views from experts on *(Digital) Twinning*. The evolving concept of *Twinning* invites us to rethink how we model, synchronize, and interact with complex systems, both physical and digital. Across the diverse domains explored in this panel — from business process management to manufacturing systems and urban DTs — it becomes clear that twinning is not a one-size-fits-all solution, but rather a paradigm requiring careful tailoring to goals, properties of interest, and operational contexts. Twinning systems must be fit-for-purpose, with clearly defined scopes and validity frames, and rely on systematic approaches for modelling and life cycle management. Twinning is also naturally dynamic: goals evolve, systems change, and models must adapt. Managing the evolution of twinning systems demands rigorous engineering practices and the correct implementation of standards for verification and validation. While technology provides the foundation, human expertise remains essential—whether in setting requirements, interpreting anomalies, or steering the continuous improvement of twins. Future research needs to further embrace hybrid approaches, combining physics-based, data-driven, and AI techniques under robust, standards-based frameworks to ensure that twinning systems remain reliable and impactful across their life cycles.

ACKNOWLEDGEMENTS

Certain commercial products and systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose. No approval or endorsement of any commercial product by NIST is intended or implied.

REFERENCES

Akcay, A., S. Biller, B. P. Gan, C. Laroque, and G. Shao. 2023. "Maintenance and Operations of Manufacturing Digital Twins". In *2023 Winter Simulation Conference (WSC)*, 1888–1899 https://doi.org/10.1109/WSC60868.2023.10407831.

Barros, A., M. Dumas, and A. H. M. ter Hofstede. 2005. "Service Interaction Patterns". In *Business Process Management*, edited by W. M. P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, 302–318. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bocciarelli, P., and A. D'Ambrogio. 2024. "Simulation-Based Predictive Process Mining with eBPMN: Methods, Challenges and Opportunities". In *2024 Annual Modeling and Simulation Conference (ANNSIM)*, 20–23 May 2024, Washington D.C., DC, USA, 1–14 https://doi.org/10.23919/ANNSIM61499.2024.10732041.

Dalibor, M., N. Jansen, B. Rumpe, D. Schmalzing, L. Wachtmeister, M. Wimmer *et al*. 2022. "A Cross-Domain Systematic Mapping Study on Software Engineering for Digital Twins". *Journal of Systems and Software* 193:111361 https://doi.org/https://doi.org/10.1016/j.jss.2022.111361.

Diakité, M., and M. K. Traoré. 2024. "Formalizing a Framework of Inference Capabilities for Digital Twin Engineering". *SIMULATION* 100(9):887–902.

El Kassis, M., P. Bocciarelli, F. Trousset, N. Daclin, A. D'Ambrogio, and G. Zacharewicz. 2024. "An HLA-based Automated Approach for the Interoperable Simulation of Collaborative Business Processes". *Simulation Modelling Practice and Theory* 135:102977 https://doi.org/https://doi.org/10.1016/j.simpat.2024.102977.

Fornari, F., I. Compagnucci, M. C. De Donato, Y. Bertrand, H. H. Beyel, E. Carrión, *et al*. 2024. "Digital Twins of Business Processes: A Research Manifesto". *Internet of Things*:101477.

Grieves, M. 2014. "Digital Twin: Manufacturing Excellence Through Virtual Factory Replication". *White paper* 1(2014):1–7.

Hua, E. Y., S. Lazarova-Molnar, and D. P. Francis. 2022. "Validation of Digital Twins: Challenges and Opportunities". In *2022 Winter Simulation Conference (WSC)*, 2900–2911. IEEE.

IEEE 2019. "IEEE-P2806: System Architecture of Digital Representation for Physical Objects in Factory Environments". IEEE.

Industry IoT Consortium 2022. "The Industrial Internet Reference Architecture". Accessed: April 5th 2024.

ISO 2003. "ISO 15926-2: 2003. Industrial Automation Systems and Integration—Integration of Lifecycle Data for Process Plants including Oil and Gas Production Facilities—Part 2: Data Model".

ISO 2015. *ISO/IEC 25024. 2015. Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Measurement of Data Quality*. ISO.

ISO 2020. *ISO/IEC TR 24028. 2020. Information Technology - Artificial Intelligence - Overview of Trustworthiness in Artificial Intelligence*. ISO.

ISO 2021a. *ISO 23247-2: 2021: Automation Systems and Integration: Digital Twin Framework for Manufacturing: Part 2: Reference Architecture*. ISO.

ISO 2021b. *ISO/IEC TR 24029-1. 2021. Artificial Intelligence - Assessment of the Robustness of Neural Networks. Part 1: Overview*. ISO.

ISO 2022. *ISO/IEC 22989. 2022. Information Technology – Artificial Intelligence - Artificial Intelligence Concepts and Terminology*. ISO.

ISO 2023. *ISO/IEC 23894. 2023. Information Technology - Artificial Intelligence - Guidance on Risk Management*. ISO.

ISO, B. 2021c. "23247-1: 2021; Automation Systems and Integration–Digital Twin Framework for Manufacturing–Part 1: Overview and General Principles". *International Organization for Standardization (ISO): Geneva, Switzerland*.

Jones, D., C. Snider, A. Nassehi, J. Yon, and B. Hicks. 2020. "Characterising the Digital Twin: a Systematic Literature Review". *CIRP Journal of Manufacturing Science and Technology* 29:36–52 https://doi.org/https://doi.org/10.1016/j.cirpj.2020.02.002.

Kang, K. C., and H. Lee. 2013. "Variability Modeling". In *Systems and Software Variability Management*, 25–42. Springer.

Lehner, D., J. Zhang, J. Pfeiffer, S. Sint, A.-K. Splettstößer, M. Wimmer *et al*. 2025. "Model-driven Engineering for Digital Twins: a Systematic Mapping Study". *Software and Systems Modeling*:1–39 https://doi.org/10.1007/s10270-025-01264-7.

Lugaresi, G., S. Gangemi, G. Gazzoni, and A. Matta. 2023. "Online Validation of Digital Twins for Manufacturing Systems". *Computers in Industry* 150:103942.

Minerva, R., G. M. Lee, and N. Crespi. 2020. "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models". *Proceedings of the IEEE* 108(10):1785–1824 https://doi.org/10.1109/JPROC.2020.2998530.

Paredis, R., and H. Vangheluwe. 2024. "Modelling and Simulation-Based Evaluation of Twinning Architectures and Their Deployment". *Proceedings of the 14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*:170–182 https://doi.org/10.5220/0012865300003758.

Plesser, H. E. 2018. "Reproducibility Vs. Replicability: A Brief History Of A Confused Terminology". *Frontiers in neuroinformatics* 11:76.

Semeraro, C., M. Lezoche, H. Panetto, and M. Dassisti. 2021. "Digital Twin Paradigm: a Systematic Literature Review". *Computers in Industry* 130:103469.

Singh, R. 1996. "International Standard ISO/IEC 12207 Software Life Cycle Processes". *Software Process Improvement and Practice* 2(1):35–50.

Tolk, A., F. J. Barros, A. D'Ambrogio, A. Rajhans, P. Mosterman, S. S. Shetty, *et al*. 2018. "Hybrid Simulation for Cyber Physical Systems: a Panel on Where Are We Going Regarding Complexity, Intelligence, and Adaptability of CPS Using Simulation.". In *SpringSim (MSCIAAS)*, 3–1.

Tolk, A., J. Lane, F. Shults, and W. Wildman. 2021. "Panel on Ethical Constraints on Validation, Verification, and Application of Simulation". In *2021 Winter Simulation Conference (WSC)*, 1–15 https://doi.org/10.1109/WSC52266.2021.9715431.

Traoré, M. K. 2023. "High-Level Architecture for Interoperable Digital Twins".

Traoré, M. K., and A. Muzy. 2006. "Capturing the Dual Relationship Between Simulation Models and Their Context". *Simulation Modelling Practice and Theory* 14(2):126–142.

UN-Habitat 2022. "World Cities Report: Envisaging the Future of Cities". *United Nations Human Settlements Programme (UN-Habitat), Nairobi, Kenya*.

Van Acker, B., P. De Meulenaere, J. Denil, Y. Durodie, A. Van Bellinghen, and K. Vanstechelman. 2019. "Valid (Re-) Use of Models-of-the-physics in Cyber-physical Systems Using Validity Frames". In *2019 Spring Simulation Conference (SpringSim)*, 1–12. IEEE.

Van der Valk, H., H. Haße, F. Möller, M. Arbter, J.-L. Henning, and B. Otto. 2020. "A Taxonomy of Digital Twins". In *AMCIS*, 1–10. Salt Lake City, USA.

Van Mierlo, S., B. J. Oakes, B. Van Acker, R. Eslampanah, J. Denil, and H. Vangheluwe. 2020. "Exploring Validity Frames in Practice". In *International Conference on Systems Modelling and Management*, June 25–26, 2020, Bergen, Norway, 131–148. Cham: Springer.

Wanasinghe, T. R., L. Wroblewski, B. K. Petersen, R. G. Gosine, L. A. James, O. De Silva, *et al*. 2020. "Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges". *IEEE Access* 8:104175–104197.

Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. 2nd ed. USA: Academic Press, Inc.

AUTHOR BIOGRAPHIES

ANDREA D'AMBROGIO is an Associate Professor at the University of Rome Tor Vergata (Italy). His research interests are in the fields of model-based systems and software engineering, dependability engineering, distributed and web-based simulation. His email address is dambro@uniroma2.it.

EDWARD Y. HUA is a Lead Modelling & Simulation Engineer at The MITRE Corporation. He received his Ph.D. from the Department of Electrical and Computer Engineering, Cornell University. His research interests include DTs, simulation and AI, and wireless networks. He is a member of ACM and INFORMS. His email address is ehua@mitre.org. His website is https://www.linkedin.com/in/edward-hua-ph-d-1296025/. His ORC-ID is 0000 − 0002 − 6294 − 3122.

GIOVANNI LUGARESI is Assistant Professor at the Department of Mechanical Engineering of KU Leuven (Belgium). He works on production planning and control, process mining, and robust optimization. His email is giovanni.lugaresi@kuleuven.be.

GUODONG SHAO is a Computer Scientist in the Systems Integration Division (SID) of the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST). He manages the Digital Twins for Advanced Manufacturing project and focuses on generic guidelines and methodologies for implementing Digital Twins and relevant standards development and testing. He is a technical expert on relevant standards committees and a co-chair of the Digital Twin Verification and Validation Working Group at the Digital Twin Consortium (DTC). His email address is gshao@nist.gov.

MAMADOU K. TRAORE is a Full Professor of System Engineering at University of Bordeaux. He holds a PhD in Computer Science from University Clermont Auvergne. His research interests lie in hybrid simulation modelling with artificial intelligence. His email address is mamadou-kaba.traore@u-bordeaux.fr.

HANS VANGHELUWE is a Professor at the University of Antwerp (Belgium) where he heads the Modelling, Simulation and Design Lab (MSDL). MSDL, one of the labs in the Antwerp Systems and Software Modelling (AnSyMo) research group, is part of Flanders Make, the Strategic Research Centre for the Flemish manufacturing industry. His research is on Multi-Paradigm Modelling (MPM) of complex, Cyber-Physical Systems (CPS) using twinning. His e-mail address is Hans.Vangheluwe@uantwerpen.be.