Engineering Digital Twins: A Research Roadmap

Benoît Combemale, Pascale Vicat-Blanc, Arnaud Blouin, Hind Bril El Haouzi, Jean-Michel Bruel, Julien Deantoni, Thierry Duval, Sébastien Gérard, Jean-Marc Jézéquel

Engineering Digital Twins: A Research Roadmap

Benoit Combemale*, Pascale Vicat-Blanc*, Arnaud Blouin†, Hind Bril El Haouzi‡, Jean-Michel Bruel§, Julien Deantoni¶, Thierry Duval‖, Sébastien Gérard**, Jean-Marc Jézéquel††

* *Inria*, France

† *INSA Rennes, Inria/IRISA, Inria*, Rennes, France

‡ *Universite de Lorraine, CRAN*, Epinal, France

§ *Université Toulouse Jean Jaurès*, Toulouse, France

¶ *Université Côte d'Azur*, Sophia Antipolis, France

‖ *Lab-STICC — IMT Atlantique*, Brest, France

** *Université Paris-Saclay, CEA, List*, Palaiseau, France

†† *Université de Rennes, Inria/IRISA, IUF*, Rennes, France

***Abstract*—As society transitions from traditional engineered physical systems to software-driven ecosystems, the demand for adaptive, cost-effective, and rapidly evolving technologies continues to rise. Digital Twins (DTs) have recently emerged as a key architectural and conceptual tool to address these needs by decoupling high-level system control and decision making from physical operations. By integrating deductive engineering models with inductive, data-driven insights, DTs offer powerful capabilities for simulation, prediction, and adaptive system management. However, despite their promise, current DT implementations often remain fragmented, domain-specific, and expensive to build and maintain. This paper proposes a research agenda for a structured and principled approach to the engineering of digital twins (EDT), grounded in established software and systems engineering practices. Our goal is to empower the creation of digital twins that are scalable, reusable, and trustworthy. Building on the capabilities outlined by the Digital Twin Consortium, we offer a complementary perspective by identifying key research challenges associated with their integration in the context of digital twin engineering. In addition, we examine the engineering lifecycle of DT systems and present a comprehensive research agenda for the EDT community.**

I. Introduction

Society is transitioning from engineered physical systems to software systems, driven by adaptability, cost-efficiency, and rapid innovation. Software allows faster, remote updates, prototyping, and global distribution, enabling a flexible, interconnected technological landscape. This shift has expanded systems from embedded to distributed Cyber-Physical Systems (CPSs), and further to Cyber-Physical Social Systems (CPSSs) or socio-technical systems when including human interaction [16]. These are applied in domains like smart cities and manufacturing, forming smart ecosystems with shared goals, such as the UN Sustainable Development Goals[1], including sustainability and ethical compliance. AI and data analytics support their monitoring and development [20]. Simultaneously, systems began incorporating explicit adaptivity management through monitoring, analysis, and planning. This gave rise to Self-Adaptive Systems (SASs) [6, 10], which use feedback loops like MAPE-K [13] to handle uncertainty, improve resilience, and optimize performance. Further development led to separating high-level control into Digital Twins (DTs) [11], layered above the physical and control systems, enabling continuous evolution and enhanced system performance [8].

Fundamentally, DTs integrate complementary knowledge: deductive models representing established engineering and physical principles with inductive models derived from operational data through machine learning. More than simple simulations, DTs strive to mirror some key characteristics of their physical counterparts — not only capturing internal states but also reflecting the system's interactions within its technical, environmental, and social context. Successfully building robust and adaptable DTs requires integrating expertise and associated stakeholders from areas like modeling & simulation, numerical analysis, data science, IoT infrastructure, networks, virtual worlds, and HCI, as well as deep domain knowledge (e.g., hydrology or fluid mechanics), etc.

Consequently, while Digital Twin technology shows considerable promise across diverse domains and the main capabilities required for their implementation have been identified [21, 4], current implementations often suffer from fragmentation, high cost, and lack of reusability.

This paper explores the foundational research, design principles, and enabling technologies needed to move beyond ad-hoc Digital Twin construction and hazardous exploitation towards a more structured, formal, and agile engineering approach. We propose key areas for future development aimed at enhancing both the efficiency, trustworthiness, and accessibility of Digital Twins through the system lifecycle.

The rest of this paper is organized as follows: we introduce the background and a motivating example in Section II, the characteristics and challenges of the DT design space are detailed in Section III, and the DT development process is analyzed in Section IV. Finally, we conclude in Section V.

II. Background and motivating example

A. Background

A *digital twin* (DT) is a software system that purposefully represents an *original*, accurately reflects its changes, can act upon its *original*, and provides added value to users or other systems.

[1] See https://sdgs.un.org/goals

The *original* refers to the real system being represented. It may take the form of a natural (physical) system (e.g., a planet or a human body), a complex anthropogenic system (e.g., a smart city), or an engineered system such as a cyber-physical system (e.g., an aircraft), a software system, a physical infrastructure (e.g., an electrical power grid), or an organizational process (e.g., a business workflow).

A DT provides, in the context of a specific purpose, an accurate and dynamic representation of the *original* system, reflecting its state and evolution in response to discrete events or at an appropriate update frequency. The accuracy and frequency are determined by the required level of fidelity in spatial and temporal dimensions, in accordance with the specific objectives of the DT implementation [1]. Similarly, a DT might act on the *original* or its environment to improve specific Key Performance Indicators (KPIs), such as sustainability, resilience, or resource efficiency [17]. Enhancing CPSSs with DTs enables deeper analysis, predictive modeling, and simulation of various scenarios. DTs also provide added value for better decision-making and performance optimization, thereby creating a bridge between the physical and digital realms [14]. They furthermore improve the modularity of SASs with a clear separation of concerns, better flexibility of the feedback loop, and advanced uncertainty management.

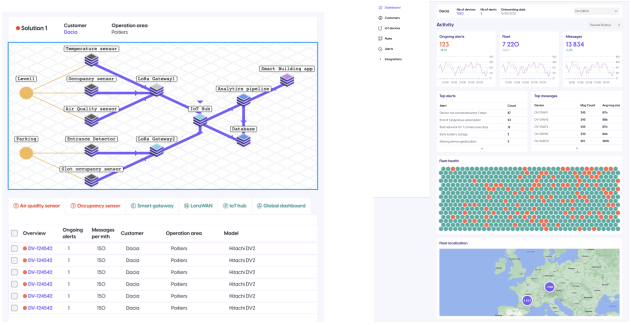
In the case of software or cyber-physical systems, one major element and difference in a number of other approaches is that we not only understand a DT as a virtual decal of an *original* during its design and development, but also as a software system that actively accompanies the *original* during its operation. In the case of a natural system serving as the *original*, the digital twin is generally constructed a posteriori, constrained by the achievable instrumentation and applicable control commands.

The DT collects, keeps, aggregates, and uses data in the form of digital shadows [5], providing users with useful information and forms of active software services, and connects with other systems. This way, the DT serves its purposes, which most likely include monitoring, analyzing, understanding, optimizing, or controlling the *original*. In reference to the object programming paradigm, we can distinguish the *DT class*, the class of DT, which contains the conceptual models (CAD, etc.) common to a range of *original systems* (see for example Figure 1a), from the DT of an *original* instance, *DT instance* which aggregates all the operational data and models attached to a real unique entity (see for example Figure 1b).

B. Motivating example

To illustrate the specifics of design space and development process challenges for DT systems and their differences from traditional software activities, we present the case of an *IoT Network* digital twin (NDT). Where relevant, we also emphasize in the following sections the specific challenges of building a digital twin when the *original* is a natural system.

At design time, such an NDT helps with network technology selection, topology design, and capacity planning, as well as financial and environmental cost estimation. During deployment, the NDT is useful for initial network device configuration decisions. During operations, it is used for performance tuning, detecting network device failures, detecting IoT traffic anomalies, simulating remediation, and automating corrective actions. It empowers on-the-field network operators (NetOps) for maintenance or evolution interventions.



(a) Conceptual models (b) Observability data

Fig. 1: How to combine design mock-ups, simulation models, and observability data?

If the network does not exist yet, the *IoT Network* designer creates a conceptual network topology with a given number of devices and assigns a theoretical configuration to each of them (see figure1a). Then, a closed-form analytical model can be used to calculate various performance and reliability metrics, or, if the network is too complex, a simulation model will be created to compare network options. For this, *IoT Network* designers define classical discrete event network simulation (NS). For example, by running what-if scenarios, they can test various topologies and capacity provisioning, and define granular radio network configuration parameters [18, 19].

At run time, the real deployed network resources are automatically discovered and listed in the device inventory of the network management server (see EQ1, EQ13). Then their status and performance are continuously monitored (see EQ5, EQ13) and reported to *IoT Network* operators via real-time dashboards (see Figure 1b). The aggregated metrics are computed and stored locally or centrally (see EQ8). To analyze these results and root cause issues (see EQ4, EQ6), the NetOps needs to leverage the data, his knowledge and the technical specifications (see EQ11). Having those available and linked via the NDT would save time (EQ13). It would also be highly relevant to benefit from real data for validating the theoretical conceptual models and, conversely, to exploit simulation results at run-time to tune real networks (see EQ2, EQ6, EQ13). However, the problem of systematically combining (see EQ1, EQ7) and comparing (see EQ3) the simulation input (eg. traffic profiles) and outputs (eg. throughput, latency) to those of real-world entities and vice versa remains open (EQ9). Collaborations between network operators and network designers could also be streamlined via the NDT (see EQ12). Interconnecting such NDT with the DT of the physical system observed (building, smart factory, etc) would help design

and operational decisions, but is still highly challenging (see EQ10).

In this example, we observe that combining design and operational tools to take full advantage of the potential of the DT raises numerous engineering questions (cf. Table I).

#	Engineering Question label	Details
EQ1	Reverse-engineering	How to transform a "running" network inventory with its associated time series to a network graph with a couple of relevant attributes that can be used for modeling and simulating the network? How to automatically map abstract network nodes of the digital model to real network nodes of the digital twin instance and vice-versa
EQ2	Model validation	How to automatically use the actual observations and performance measurements of a deployed network to calibrate and validate the network model?
EQ3	Data consolidation	How to compare the actual measured time series with the simulation results? How to associate context and quality score with raw data and simulation results?
EQ4	Decision support	If there is a need to expand the topology (add a new IoT device) or to change the periodicity of the IoT data collection, the NetOps would like to use its DT to simulate the future scenario and check its validity. For this, he would like to run the network simulator with the model corresponding to the future network, with their configuration parameters and exact geolocation, as physical characteristics strongly impact radio propagation. How to translate this automatically in the network model taken as input in the simulation?
EQ5	Digital coupling abstraction	How to specify and abstract the properties of the DT connectivity and data processing to hide the technical implementation details and compare alternatives for sensor placement, data sampling rate selection, and data preparation adapted to each DT service? What to do when data are not well collected?
EQ6	Model hybridation abstraction	In this example, we see the decoupling between the software components developed for decisions to be made at design-time (network design, modeling, and simulation) and those developed for decisions to be made at run-time (network and traffic monitoring, data analytics and machine-learning capabilities). We also understand the potential of combining them, but what are the artefacts to achieve this, not only for this specific IoT network domain, but also for any other domain?
EQ7	Development process	The building of a DT for an IoT network can start by integrating the tools developed for runtime or design. To capitalize on these investments and get the benefits of them throughout the life of the network, it would be beneficial to integrate and combine these software. What are the constraints? what is the process?
EQ8	Deployment considerations	The various components of the DT can be distributed between network devices and servers. How to ensure that they all run on the same software version or that the DT is informed of discrepancies.
EQ9	Continuous consistency	How to maintain the consistency between the conceptual simulation model and the real network when it evolves. It would be nice to have automated updates
EQ10	Federation	How to combine the respective digital twin of two interconnected networks? How to leverage their simulation results as well as their respective data, given the fact that the two networks exchange traffic bidirectionally.
EQ11	Technical specs ingestion	Manufacturers' technical data sheet of real equipments contains a lot of information (technical bounds, etc.). How can we ingest and exploit this data for both AI and simulation validation?
EQ12	Collaboration	Network designers and network operators would benefit from each other. An annotation system would be an interesting feature for this.
EQ13	Model, data, what-if scenario and results catalogs	Storing and retrieving easily models and associated data or results, as well as scenario or context, is of paramount importance for understanding results and recommendations. As the DT collect, aggregate but also generate a lot of data, there is an important need to facilitate the navigation in this complex information space.

TABLE I: Examples of DT Engineering Questions

III. Design Space

The development of DTs requires an intricate blend of technologies spanning multiple scientific and engineering fields. The DTC's Periodic Table of Digital Twin Capabilities [9] provides a structured taxonomy of the core building blocks that can be integrated while developing a DT. Organized into capability domains – data services, intelligence and cognition, user experience, management, trustworthiness, and integration – the table serves as a reference framework for understanding the diverse functional components needed to build a DT. However, while it effectively outlines the various possible building blocks to be considered, it does not address how these building blocks must be integrated into a coherent DT that meets the requirements. As a complex software system that integrates diverse, heterogeneous, and interdependent capabilities in a cohesive and synergetic manner, *software and systems engineering* plays a pivotal role in the engineering of DT [3, 15]. The engineering of such systems requires rigorous foundations, robust toolchains, and well-defined methodologies to manage their inherent complexity. This is essential to ensure the scalability, reliability, reusability, and maintainability of DTs throughout their life cycles (incl. development, deployment, and continuous evolution).

To fully realize the potential of DTs, a comprehensive research agenda is required to establish foundational principles and a unified development framework. Although current DT initiatives provide valuable insight and important domain-specific building blocks, there remains a critical need for a structured and formal engineering approach to DT. In the following, we focus on the engineering challenges that arise when combining these heterogeneous capabilities in real-world implementations. We examine engineering digital twin research questions (denoted as **RQ_Xi**) raised in the various dimensions **X** of the design space, following the typology of capabilities of the periodic table from the DTC.

A. Data Services

Data management (**D**) is a foundational pillar of DT systems, as it directly determines the quality, reliability, and adaptability of the digital representation. In the context of DT engineering, it must address the acquisition, selection, transformation, storage, fusion, and long-term governance of data originating from heterogeneous and dynamic sources – including sensors, simulations, human expertise, and external services. One of the first scientific challenges lies in defining *what* data to collect, *when and where*, and *which sensing technologies and communication strategies* to employ. This selection must be guided by the nature of the *original*, the variability of its environment, and above all, the constraints of the targeted DT services (e.g., real-time diagnosis, predictive maintenance, simulation).

RQ_D1: *Where and when should sensors be deployed to extract relevant information, depending on the DT service constraints? What types of sensors and communication technologies are most suitable to support efficient, adaptive, and scalable data acquisition? (see EQ5)*

In many application domains, certain key variables are not directly measurable or are only partially observable in time and space. It is essential to explore reconstruction strategies, such as inverse methods, observer-based estimation, or learning-based imputation, in order to infer missing or hidden information and support robust decision-making.

RQ_D2: *How to address unmeasurable or partially observable data (in time and space) in dynamic systems? What specific inverse methods or reconstruction techniques can be developed to estimate missing information and support decision making under incomplete and uncertain observations? (see EQ5)*

These aforementioned data-centric challenges are part of a broader objective: to build a comprehensive and robust framework for intelligent data use. This includes not only acquisition and inference, but also pre-processing, quality assurance, fusion of multi-modal sources, and integration into the DT lifecycle.

RQ_D3: *How can we design a comprehensive framework for the robust and continuous collection, processing, and intelligent use of heterogeneous, noisy, and incomplete data? (see EQ5)*

It is also important to reason about data originating from diverse sources—such as measurements, simulation models, and prediction models. To automate this multi-source reasoning, all data should be augmented with metadata that enables the detection of potential reasoning. Furthermore, these metadata should be managed seamlessly, regardless of their origin.

RQ_D4: *What are the appropriate metadata to ensure detection of synergy between data from different models and their potential representative monitored in the original? (see EQ3 and EQ6)*

RQ_D5: *How to make the metadata available and their manipulation homogeneous? (see EQ3 and EQ13)*

When synergies are identified between data from different models, it is important to ensure that these synergies can be easily exploited, potentially by modifying the models and their parameters or by creating a new one. For example, this could enable autonomous and continuous calibration of a simulation model against original monitoring data, or facilitate autonomous and continuous training of an AI model.

RQ_D6: *What are the appropriate architectural patterns that enable safe and continuous evolution (or creation/suppression) of simulation models wrt. observations from the original?* [2] (see EQ2, EQ3 and EQ9)

RQ_D7: *What would be a simulation model manipulation interface that would allow CRUD-like manipulations on heterogeneous models in a homogeneous and comprehensive way?* (see EQ1, EQ3, EQ9 and EQ12)

RQ_D8: *What are the appropriate architectural patterns to enable safe and continuous training, tuning (or creation) of AI models wrt. observations from the original?* (see EQ1, EQ3, EQ4 and EQ9)

RQ_D9: *What would be an AI model manipulation interface allowing CRUD-like manipulations over heterogeneous models in a homogeneous and comprehensive way?* (see EQ1, EQ3, EQ9 and EQ12)

When the time comes for federating or composing different digital twins, alignment between the data (resp. metadata) of the various DTs is of utmost importance, raising new challenges.

RQ_D10: *When a DT is composed of a set of sub-DTs, how can we effectively aggregate their data feeds, e.g., high-level monitoring or scalable machine learning?* (see EQ7)

RQ_D11: *How can we effectively tackle semantic interoperability across federations of DTs?* (see EQ10)

B. Intelligence and Cognition

Intelligence and cognition (**I**), through model management are essential elements for a DT to provide reliable insights. Beyond specific usage of models in a DT, there should be a systematic way to account for the appropriate usage of the different models, for instance ensuring the relevant accuracy of a model to take specific decisions.

RQ_I1: *How to specify the conditions under which a deductive model can be consistently used/substituted for a specific usage? (e.g., in terms of its validity envelope/frame)* (see EQ2, EQ3 and EQ13)

RQ_I2: *How to specify the conditions under which an inductive model can be consistently used/substituted for a specific usage? (e.g., in terms of its FATES+ properties)* (see EQ11 and EQ13)

Furthermore, hybridization between deductive and inductive models is now a recognized practice for improving specific properties of models (e.g., reducing their uncertainty or expanding the domain of use). However, it is usually done in a handcrafted and ad-hoc way [7].

RQ_I3: *What are the patterns used to leverage the hybridization of deductive and inductive models?* (see EQ6, EQ9 and EQ12)

RQ_I4: *How can we provide composition operators to apply hybridization patterns?* (see EQ6, EQ9 and EQ12)

RQ_I5: *How does a hybridization pattern affect the validity frame and/or the FATES+ properties of a hybrid model? More generally how does it affect its metadata?* (see EQ2, EQ3, EQ6, EQ9, EQ12 and EQ13)

C. User Experience

By its very nature, a DT strongly relies on data visualization and user interactions (**U**). Indeed, a DT usually embeds interactive systems that enable users to visualize and interact with both physical and DTs, and their large amount of associated data [17]. Those interactive systems can take multiple forms, from fully digital virtual reality (VR), immersive systems to hybrid augmented reality (AR) systems [23], but also 2D graphical user interfaces, and systems that involve physical input devices. They often propose collaborative asymmetrical (AV/VR/2D) interactive situations, which also raise challenges [22, 12]. Building such

interactive systems from scratch or without automation is time-consuming and may prevent the generalization of DTs as industrial tools. A first research challenge concerns the creation of the visual representation of the DT.

RQ_U1 *How to ease and accelerate the creation of DTs' visual representations from existing descriptive artefacts?* (see EQ13)

Visualization techniques play a crucial role. A specific point of DT is the uncertainty of the data acquired from the original and displayed within the DT.

RQ_U2 *How to operate within a DT to enhance the perception and understanding of the* original*? (see EQ13)*

Input devices within DT have specific challenges. In particular, it is challenging to design and develop input devices and their interactions to enable stakeholders to control DTs.

RQ_U3: *How to design and program DT's input devices and their interactions? (see EQ7)*

Another specificity of DTs is not only their ability to navigate in space but also to navigate in time.

RQ_U4: *How to manage variations in space and time within a DT? (see EQ13)*

DTs have specific challenges related to usability. In particular, we need to find usable ways to give DT stakeholders accountability for their actions, as a DT may control an original.

RQ_U5: *How to evaluate and validate the usability of DTs? (see EQ13)*

Collaboration within DTs implies specific challenges. This concerns the social coordination between the different stakeholders of a DT to share results. This also requires specific conversation mechanisms in relation to the original.

RQ_U6: *How to collaborate within a DT in relation to the original?(see EQ11)*

Moreover, the federation and composition of DT question the way stakeholders can interact in collaboration with them.

RQ_U7: *When a DT is composed of a set of sub-DTs, how can we effectively aggregate their UI?(see EQ10)*

D. Management

Connectivity between the physical system and its DT is a key enabler of continuous synchronization, real-time observability, and responsive decision making. Managing and controlling this connectivity (**C**) to ensure reliable, efficient, and adaptive communication between these two entities raises several scientific and technical challenges, especially when operating in dynamic, constrained, or mission-critical environments.

RQ_C1: *How to dynamically orchestrate connectivity strategies in a Cloud–Edge–IoT continuum, based on usage priorities, context variations, and resource constraints?*

RQ_C2 : *What adaptive mechanisms can ensure seamless reconfiguration and resilience of DT-to-physical-system links in dynamic and mobile contexts?*

RQ_C3: *How can we formalize and implement co-optimization strategies between the network and the DT to balance communication constraints and service requirements dynamically?*

RQ_C4: *How to design bidirectional data exchange patterns that ensure time-coherent, safe, and explainable feedback loops between the physical and digital layers?*

RQ_C5: *When a DT is composed of a set of sub-DTs, how can we effectively coordinate and synchronize their need to access their physical twins?* (see EQ10)

Finally, **frugality and resilience** are key design goals. Not all data should be transmitted at high frequency or centralized. Strategies such as local filtering, compression, prioritization, or event-driven communication can drastically reduce energy consumption and improve responsiveness, especially in embedded or edge environments (see EQ3)

RQ_C6: *How to balance communication cost, energy usage, and data quality in the design of adaptive and frugal connectivity infrastructures for DTs?* (see EQ3?)

E. Trustworthiness

In the context of the engineering of DTs, Trustworthiness (**T**) refers to the quality attributes of all the capabilities. This can include aspects such as fidelity, accuracy, performance, uncertainty, and consistency of the model/data. The data used in a DT, depending on its source and underlying technologies, has different quality properties. Such properties may affect the quality properties of the consumers of these data, and it is important to understand how.

RQ_T1: *How do data quality properties affect the properties/metadata of models trained or calibrated on such data?(see EQ1, EQ3, and EQ9)*

More generally, it is important to exhibit the quality properties of the various models used in a DT (e.g., in their metadata). To this end, various analysis techniques must be provided to compute the quality properties of the model and their possible hybridization.

RQ_T2: *What are the appropriate analysis techniques to automatically obtain a model quality properties? Does it require access to the model or can it be done in a black box manner?*

RQ_T3:*When a DT is composed of a set of sub-DTs, how can we effectively compound uncertainty, scale, fidelity, and assumptions that stem from the individual DTs? (see EQ7 and EQ10)*

RQ_T4: *How can we effectively quantify and validate the accuracy and fidelity of DTs? What are the best methods for calibrating these digital representations to ensure they accurately mirror the physical system? (see EQ2?)*

RQ_T5: *Which properties are important for each expected use cases of DTs in order to assess the quality of this latter?*

RQ_T6: *How these quality properties can be continuously monitored and improved?*

Finally, DTs should adhere to standardized and regulated quality attributes. This raise the need for ongoing support and continuous assessment to ensure compliance with these standards and regulations.

RQ_T7: *How can DTs be designed to comply with relevant regulations and standards?*

F. Integration

While it is usually the case that a DT and the underlying models and services provide functional interfaces, it is also important to allow access to the underlying metadata, enabling conscious use of the models and services. This is key to integrate a DT in its ecosystem (**E**) [8].

RQ_E1: *Beyond functional interfaces, what is the appropriate interface to manipulate and query the metadata associated with the heterogeneous models and services?* (see EQ10)

RQ_E2: *How can we modularize DTs to enable flexible composition and address challenges related to variability management, trustworthiness, economics, and ethics?* (see EQ7).

RQ_E3: *How to dynamically compose or orchestrate DTs at the deployment level to federate and coordinate DTs?* (see EQ10)

RQ_E4: *How can we effectively tackle semantic interoperability across various perspectives/viewpoints needed in engineering DTs?* (see EQ3)

RQ_E5: *How to ensure data security and traceability?* (see EQ10)

RQ_E6: *What standards and protocols are essential for ensuring interoperability between different Digital Twin (DT) components, and how can industry standards (e.g., ISO, IEC) be applied to enhance integration in engineering DTs?* (see EQ10)

IV. Process

The DT engineering activities are not bound to a traditional linear or circular DevOps timeline. They revolve around the continuous, consistent integration and linkage of independent building blocks and the manipulation of a dynamic graph of data and models that glue them. A DT can be created at different points in the lifetime of its real counterpart. It can be created 1) **after** the *original system*, 2) **before** the *original system* exists, or 3) **at the same time** of the *original system* (co-design). In case 1), the original (or its environment) must be instrumented with sensors and actuators, a data pipeline set in place, and some data analytics capabilities deployed. This corresponds to *brownfield engineering* [5]. In this case, and according to the kind of originals, the DT models can be reverse-engineered from the instrumented original that is intended to mirror, built via machine learning, or reused from the design of the original itself.

In case 2), it may be possible that a design model (CAD), an analytical model, a simulation or a descriptive model already exists and can be used to produce initial data and models to populate the DT. If no digital model of the to-be-built original already exists, the DT is developed from scratch.

In case 3), the digital model is created while a prototype of the original is created and instrumented, a data pipeline set in place. Case 3) corresponds to *greenfield engineering* (both systems are conceived and developed together) [5].

These different cases, together with the kind of original under consideration (natural, anthropized or engineered system), impact the engineering processes and the constraints to consider when developing the digital twin.

We list below a set of research questions related to the DT engineering process :

RQ_P1: *What are the viewpoints and model kinds the different stakeholders need to address their concerns? What architectural framework for engineering digital twins?*

RQ_P2: *What communication tools and practices can enhance collaboration between data scientists, engineers, and operations teams in a DT engineering environment? (see EQ11)*

RQ_P3: *What communication strategies can ensure that all stakeholders are informed and engaged throughout the DT models lifecycle? What collaboration tools and platforms are most effective for facilitating teamwork in managing models for digital twins? (see EQ11)*

RQ_P5: *What are the industry-specific challenges and solutions for registering and managing models and data in a repository for digital twins in different sectors (e.g., manufacturing, healthcare, energy)? (see EQ13). What emerging technologies (e.g., federated learning, and MPC) can enhance this registration and management? (see EQ8)*

RQ_P6: *What key factors should be considered when scaling Digital Twin systems to accommodate large volumes of data and intricate engineering applications?*

RQ_P8: *What are the best practices for deploying and versioning models in digital twin systems to ensure seamless integration and operation? (see EQ5?)*

RQ_P9: *How can security be integrated into the DevOps pipeline (DevSecOps) to ensure the protection of digital twin systems and associated data?*

RQ_P10: *What steps can organizations take to ensure compliance with relevant regulations and standards when integrating Digital Twin systems with engineering applications?*

V. Conclusion

In this paper, we emphasize the need for a more structured and principled engineering approach to Digital Twin (DT) development. While DT technology holds significant promise across various domains, its current applications are often fragmented and costly. We propose a comprehensive research agenda, focusing on key areas of DT capabilities such as data and model management, intelligence, integration, user experience, and trustworthiness. A unified framework for DT engineering is necessary to ensure scalability, reliability, and continuous evolution. The research agenda includes addressing data quality, the integration of inductive and deductive models, and improving collaboration across different stakeholders. To fully leverage the potential of DTs, significant efforts are required to design adaptable, resilient systems that align with evolving needs. This research agenda aims to drive future advancements, making DTs more accessible, reliable, and effective across industry, science, and public institutions.

References

[1] Emmanuelle Abisset-Chavanne, Thierry Coupaye, Fahad R. Golra, Damien Lamy, Ariane Piel, Olivier Scart, and Pascale Vicat-Blanc. A Digital Twin use cases classification and definition framework based on Industrial feedback. *Computers in Industry*, 161:104113, 2024.

[2] Tarek Alskaif, Önder Babur, Francis Bordeleau, Loek Cleophas, Benoît Combemale, Joachim Denil, Øystein Haugen, Judith Michael, Phu H. Nguyen, Tiberiu Seceleanu, Mark van den Brand, and Hans Vangheluwe. Evolution at the Core of Digital Twin Engineering. In *Proceedings of the 2nd International Conference on Engineering Digital Twins*, Grand Rapids, Michigan, United States, 2025.

[3] Francis Bordeleau, Benoît Combemale, Romina Eramo, Mark van den Brand, and Manuel Wimmer. Towards model-driven digital twin engineering: Current opportunities and future challenges. In Önder Babur, Joachim Denil, and Birgit Vogel-Heuser, editors, *Systems Modelling and Management - First International Conference, ICSMM 2020, Bergen, Norway, June 25-26, 2020, Proceedings*, volume 1262 of *Communications in Computer and Information Science*, pages 43–54. Springer, 2020.

[4] Hugh Boyes and Tim Watson. Digital twins: An analysis framework and open issues. *Computers in Industry*, 143:103763, December 2022.

[5] Stefan et al. Braun. Engineering Digital Twins and Digital Shadows as Key Enablers for Industry 4.0. In *Digital Transformation*, pages 3–31. Springer Berlin Heidelberg, Berlin, Heidelberg, 2023.

[6] Betty H. C. Cheng and al. *Software Engineering for Self-Adaptive Systems: A Research Roadmap*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[7] Benoit Combemale, Jeff Gray, and Bernhard Rumpe. Model hybridization: towards a unifying theory for inductive and deductive reasoning. *Softw. Syst. Model.*, 23(6):1307–1308, December 2024.

[8] Benoît Combemale, Jörg Kienzle, Gunter Mussbacher, Pascal Archambault, Jean-Michel Bruel, Loli Burgueño, Betty H C Cheng, Loek Cleophas, Gregor Engels, Damien Foures, Stefan Klikovits, Vinay Kulkarni, Judith Michael, Sébastien Mosser, Houari Sahraoui, Eugene Syriani, and Andreas Wortmann. On the Challenges of Integrating Digital Twins. In *Proceedings of the 2nd International Conference on Engineering Digital Twins*, Grand Rapids, Michigan, United States, 2025.

[9] Digital Twin Consortium. Digital Twin Capabilities Periodic Table (v.1.2), 2025. https://www.digitaltwinconsortium.org/initiatives/capabilities-periodic-table/ [Accessed: 2025-07-04].

[10] Rogério de Lemos and al. *Software Engineering for Self-Adaptive Systems: A Second Research Roadmap*, pages 1–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

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

[12] Arthur Fages, Cédric Fleury, and Theophanis Tsandilas. Understanding Multi-View Collaboration between Augmented Reality and Remote Desktop Users. *Proceedings of the ACM on Human-Computer Interaction* , (549):27 pages, November 2022.

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

[14] Kendrik Yan Hong Lim, Pai Zheng, and Chun-Hsien Chen. A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. *J. Intell. Manuf.*, 31(6):1313–1337, August 2020.

[15] Judith Michael, Loek Cleophas, Steffen Zschaler, Tony Clark, Benoit Combemale, Thomas Godfrey, Djamel Eddine Khelladi, Vinay Kulkarni, Daniel Lehner, Bernhard Rumpe, Manuel Wimmer, Andreas Wortmann, Shaukat Ali, Balbir Barn, Ion Barosan, Nelly Bencomo, Francis Bordeleau, Georg Grossmann, Gabor Karsai, Oliver Kopp, Bernhard Mitschang, Paula Muñoz Ariza, Alfonso Pierantonio, Fiona A. C. Polack, Matthias Riebisch, Holger Schlingloff, Markus Stumptner, Antonio Vallecillo, Mark van den Brand, and Hans Vangheluwe. Model-driven engineering for digital twins: Opportunities and challenges. *Systems Engineering*, n/a(n/a).

[16] Regina Reine, Filbert H. Juwono, Zee Ang Sim, and W. K. Wong. *Cyber-Physical-Social Systems: An Overview*, pages 25–45. Springer International Publishing, Cham, 2021.

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

[18] Samir Si-Mohammed, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. HINTS: A methodology for IoT network technology and configuration decision. *Internet of Things*, 22:100678, 2023.

[19] Samir Si-Mohammed, Zakaria Fraoui, Thomas Begin, Isabelle Guérin Lassous, and Pascale Vicat-Blanc. Stacknet: Iot network simulation as a service. In *IEEE International Conference on Communications (ICC 2023)*, 2023.

[20] Aakash Singh, Anurag Kanaujia, Vivek Kumar Singh, and Ricardo Vinuesa. Artificial intelligence for Sustainable Development Goals: Bibliometric patterns and concept evolution trajectories. *Sustainable Development*, 32(1):724–754, February 2024.

[21] Pieter Van Schalkwyk, Sean Whiteley, and Marc) Goldman. Digital Twin Capabilities Periodic Table User Guide, Version 1.2, May 2025.

[22] Chiu-Hsuan Wang, Chia-En Tsai, Seraphina Yong, and Liwei Chan. Slice of light: Transparent and integrative transition among realities in a multi-hmd-user environment. In *Proceedings of the 33rd Annual ACM*

Symposium on User Interface Software and Technology, UIST '20, page 805–817, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. Embedded data representations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):461–470, 2017.