From modeling and simulation to digital twin : evolution or revolution?

Reference:
Ali Zeeshan, Biglari Raheleh, Denil Joachim, Mertens Joost, Poursoltan Milad, Traoré Mamadou Kaba.- From modeling and simulation to digital twin : evolution or revolution?
Simulation - ISSN 1741-3133 - 100:7(2024), p. 751-769
Full text (Publisher's DOI): https://doi.org/10.1177/00375497241234680
To cite this reference: https://hdl.handle.net/10067/2045340151162165141

Journal Title
XX(X):1–17
©The Author(s) 2023
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

From Modeling and Simulation to Digital Twin: Evolution or Revolution?

Zeeshan Ali[1], Raheleh Biglari[2], Joachim Denil[2], Joost Mertens[2], Milad Poursoltan[1], and Mamadou Kaba Traoré[1]

Abstract

As digitalization is permeating all sectors of society towards the concept of ”smart everything”, and virtual technologies and data are gaining a dominant place in the engineering and control of intelligent systems, the Digital Twin (DT) concept has surfaced as one of the top technologies to adopt. This paper discusses the DT concept from the viewpoint of Modeling and Simulation (M&S) experts. It both provides literature review elements and adopts a commentary-driven approach. We first examine the DT from a historical perspective, tracing the historical development of M&S from its roots in computational experiments to its applications in various fields and the birth of DT-related and allied concepts. We then approach DTs as an evolution of M&S, acknowledging the overlap in these different concepts. We also look at the M&S workflow and its evolution toward a DT workflow from a software engineering perspective, highlighting significant changes. Finally, we look at new challenges and requirements DTs entail, potentially leading to a revolutionary shift in M&S practices. In this way, we hope to foster the discussion on DTs and provide the M&S expert with innovative perspectives.

Keywords

Modeling and Simulation, Digital Twin, Evolution, Revolution, Workflow

1 Introduction

Modeling and Simulation (M&S) is an established scientific discipline. Computer simulation has its roots in the computational experiments of neutron scattering developed by Ulam and Von Neumann, who developed the Monte Carlo method[1,2]. Since then, researchers have made many foundational contributions to M&S. The number of applications of simulations to solve real-world problems is even many times more. Computational science has even been called the third pillar of science for that reason[3].

In recent years, the Digital Twin (DT) concept has surfaced in many areas e.g., aerospace[4], manufacturing[5], healthcare[6], transportation systems[7] and smart cities[8]. The DT approach landed in the top strategic technology trends, as shown in the Gartner hype cycle of 2017[9] and 2018[10]. However, for most M&S researchers, the DT concept seems like a natural evolution of M&S. Though seemingly evolutionary, there might be some revolutionary aspects to the concept. In this paper, we look at the DT from both perspectives and identify challenges related to the DT from the perspective of an M&S practitioner.

The various fields and industries that employ DTs all have their own understanding of the concept influenced by the way they use it to create more value in their business. This has, unfortunately, resulted in a plethora of definitions[11] of the term, which has hollowed it out. We therefore briefly revisit five historical developments from the classic simulation model to the DT concept. We also look at the current viewpoints of some major industrial players and standardization bodies, as well as recent classifications suggested in the literature.

This paper discusses the transitioning from M&S to DT. First, section 2 discusses a historical overview of digital twins. In section 3, we discuss how the current state of digital twinning is an evolution from M&S; the evolutionary process of the M&S workflow towards a DT workflow is presented. In section 4, we discuss how DT can be seen as a revolution due to some drastic changes in traditional M&S practice. New requirements for the adoption of DT are paired with open challenges, which we discuss. Finally, we conclude the paper in section 5 and give some perspectives.

2 Digital Twin: a Historical Perspective

In the 1960s the DT idea was born at NASA from the “living model” of its Apollo missions[12]. Figure 1 illustrates the historical perspective of the emergence and budding of the DT concept.

The evolution of M&S has progressed alongside the significant advancements witnessed in computer science. This is why the history of M&S dates back to the 1960s when the first computers could support simulations. During the initial decades of simulation development, the generated output primarily consisted of textual reports[13].

[1]University of Bordeaux, France
[2]Flanders Make @UAntwerpen, University of Antwerp, Belgium

Corresponding author:
Mamadou Kaba Traoré, University of Bordeaux IMS Laboratory, CNRS UMR 5218, 351 Cours de la Libération, Bat. A31, 33405, Talence, France.
Email: mamadou-kaba.traore@u-bordeaux.fr

In the 1980s, visual animation was integrated into commercial simulations to enable all stakeholders to communicate with the model appropriately. Since the 1980s, real-time simulation (RtS) has been increasingly used in industrial and entertainment applications[14]. In this approach, the simulation model generates outputs and responds to inputs at a pace corresponding to the system's real-world dynamics. These developments influenced other industries to adopt real-time simulation techniques for training, planning, and testing purposes. RtS possesses the capability not only to model the future state of systems but also to depict the current state of real systems such as hospitals, factories, distribution hubs, supply chain networks, airports, and container ports. The term "RtS" can be traced back to the 1950s, when Rubinoff defined RtS as a simulation of the performance of a process or device at its regular operating speed. The digital simulator is responsible for keeping up with the simulated process or device[15]. However, the RtS concept has evolved over time. The MIT Servomechanisms Laboratory and the United States Air Force developed the Whirlwind I project, which was publicly announced in 1951[16]. Historically, the project has consistently focused on the domains of real-time simulation and control[17]. Most RtSs are identified by including some physical components in the simulation. This might be hardware, software, or a human(s) in the loop. In every case, the simulation needs to be synchronous with a wall clock to ensure the correct timing of the interactions between the simulation and the external agent[18].

Simulation is one of the most important tools for evaluating system performance and assisting in decision-making. However, RtS for decision-making plays a key role in many sectors, spanning from manufacturing plants[19,20] to sociotechnical systems such as healthcare services[21].

In 1991, we find Gelernter's concept of "Mirror Worlds", a software model of a part of the real world, fed with information streams such that the model is ever up-to-date with reality. He envisions this mirror world to be accessible by multiple users simultaneously, each of which can request and view exactly those aspects of the model they are interested in, at whatever level of detail necessary. A city Mirror World would contain the state of bridges, locations of policemen, occupancy of buildings, etc. Those occupied buildings would have a mirror world themselves, with, for example in a hospital, digital versions of patients and doctors, but also rooms and medical inventory[22]. Gelernter's view is that of a doppelgänger and is bordering on what is nowadays called a DT.

The origin of the Symbiotic Simulation can be traced back to 1998 to Davis' concept of online simulation[23], but the term Symbiotic was introduced at the Dagstuhl seminar on Grand Challenges for Modeling and Simulation in 2002[24].

A Symbiotic Simulation System is a system in which a simulation and physical system interact with each other through an exchange of data. The physical system sends measurements to the simulation, which sets up what-if experiments to control or influence that physical system optimally. In this initial definition, the Symbiotic Simulation System forms a closed loop. It behaves in a mutually beneficial way, but this does not need to be the case, as argued by Aydt et al.[25]. If the model used in the simulation does not accurately represent the physical system, suboptimal or even detrimental decisions might be made. Aydt et al.[25] also propose other uses of the Symbiotic Simulation System that do not require a closed loop, such as forecasting, model validation, and anomaly detection. We see that Symbiotic Simulation Systems have found their way into Industry 4.0[26], and as Cao et al. put it succinctly[27]: "Symbiotic simulation systems describe the whole process of using a DT...".

Grieves's initial concept of DT is found in a University of Michigan presentation on Product Lifecycle Management (PLM). At the time, the slide was called "Conceptual Ideal for PLM", and over time, the concept was renamed to "the information mirroring model" initially and later on to "digital twin". This slide, reproduced in Figure 2, contains all the elements of what is nowadays considered a DT: a real space, which is mirrored by a virtual space (VS) that consists of any number of sub-spaces (VS1, VS2, ..., VSn), and the accompanying data flow from the real space to the virtual one, as well as an information flow in the opposite direction. The central idea is that the real space represents a physical system, and the virtual space represents all the information of this physical system throughout its lifecycle, from (prototype) production to disposal[28]. This view is similar to Gelernter's concept but brought in the more specific context of PLM, and with the extension of an automated information flow from the virtual to the real space. It is also clear that there is definite overlap with Symbiotic Simulation Systems. The main difference is that in Symbiotic Simulation, the one-to-one mapping of real space to virtual space is not a necessity, though it may be optionally present, e.g. in its model validation usage.

2010 brings us NASA's definition, which stems from a roadmap on modeling, simulation, and information technology[29]. NASA describes a DT as follows: "A Digital Twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin"[29,30]. They consider the DT to be ultra-realistic, based on high-fidelity physical models, onboard sensor data, maintenance history, and fleet data. The central idea is that due to future missions being more complex and longer, a DT can aid by continuously forecasting the system health and the probability of mission success, as well as uncover issues before they become critical[30]. Compared to Grieves' view, this definition is less general and clearly influenced by an aeronautics background, yet the basic elements from Figure 2 remain present.

Several industrial consortia, individual researchers, and standardization bodies are actively participating in the progress of DT technology. For example, "Alliance Industrie du Future" (AIF, a large French consortium of industries and academics) defines the DT as (i) an organized set of digital models representing a real-world entity designed to address specific issues and uses, (ii) updated in relation to reality, with a frequency and precision adapted to its issues and uses and, (iii) equipped with advanced operating tools including the ability to understand, analyze, predict and optimize the operations and management of the real entity[31]. As stated by the Digital Twin Consortium,

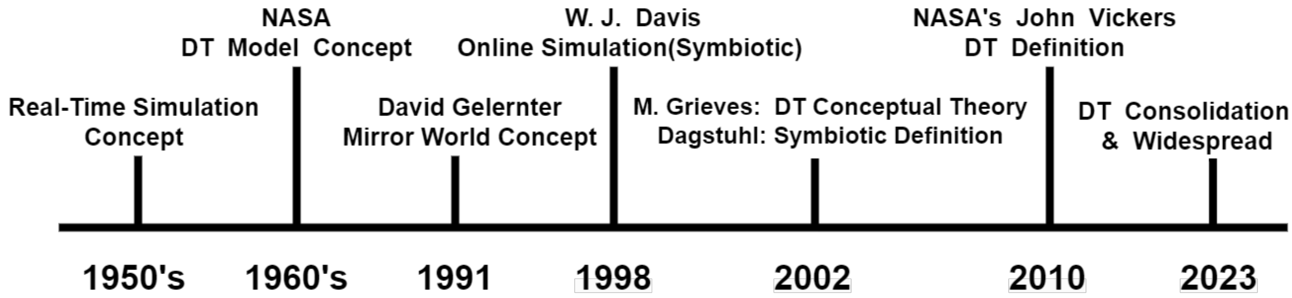

Figure 1. Historical perspective of the Digital Twin concept

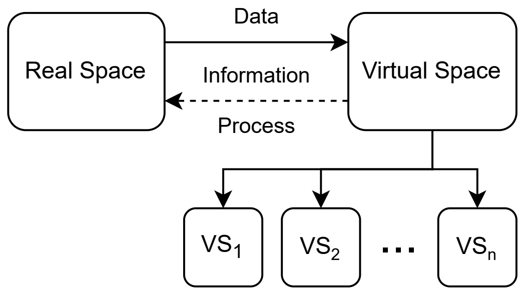

Figure 2. Adaptation of Dr. Michael Grieves' slide from 2002[28].

a DT is a virtual replication of a real-world physical object, entity, or process, which is synchronized with its physical counterpart at a certain frequency and fidelity. The DT supports businesses through holistic understandings, optimal decision-making, and effective actions. DT predicts possible future scenarios and represents the present and past, by using real-time and historical data[32]. According to Siemens, a DT is a virtual duplicate of an object, machine, process or a complete facility of production. It can carry all the data and models relevant to the real-world entity along all the value chain processes, from design to production, operation, maintenance, and the recycling of the product. This makes it possible to design, simulate and manufacture products faster whilst improving the factors of economy, performance, robustness or environmental compatibility[33]. General Electric (GE) defines the DT as a software representation of a physical asset, system, or process designed to detect, prevent, predict, and optimize through real-time analytics to deliver business value[34]. In an attempt to consolidate various DT definitions, Wright[35] distilled the following three required parts in a DT: (i) a model of the twinned system is needed, (ii) an ever-evolving dataset related to the twinned system is needed, (iii) a means of updating the model in accordance with the data is needed. This aligns with the definition given by the ISO (International Organization for Standardization), which defines the DT in a manufacturing context as the digital representation of an observable manufacturing element with synchronization between the element and its digital representation[36].

Because all these definitions are rather broad, there have been attempts in the literature to define further classifications in the DT technology according to some properties. Kritzinger et al.'s classification[5] is a classification based on the level of data integration between a physical object and its digital counterpart. Specifically, the classification looks at the presence of computer-automated data/information exchange between the digital world and the real-world entity, as shown in Figure 3. Based on that presence, it distinguishes between a digital model, digital shadow, or digital twin. A digital model only has manual data exchange between the two objects, a digital shadow has automatic data exchange from the physical object to the digital one, and a digital twin has an automated data exchange in both directions; that is, the digital object can directly influence the physical one. Despite stemming from the manufacturing field, this classification is generic and broadly applicable. Babic[37] classifies DTs for smart manufacturing in two groups based on the awareness of the digital twin about the manufacturing equipment's layout. He classifies them as the static twin, in which the equipment layout is configured manually, and dynamic, in which the twin automatically determines this configuration. Bao et al.'s classification[7] also stems from manufacturing, yet their classification focuses on what the DT captures: the produced product or the production process. They describe the product DT as a virtual information carrier of the product that carries information associated with that product through the various phases of its life, that is, design, manufacturing, maintenance, repair, and operations. The process DT then supports the production process and captures the appropriate attributes and manufacturing procedures in a digital way.

3 Digital Twin: An Evolution?

Some scholars posit that DT can be attributed to the evolution of the simulation model. For example, Lugaresi et al.[38] state that DT can be seen as an evolved form of simulation that has been used for years. Compared to mere simulation, the evolved features are a bidirectional data flow and synchronization between the real and the virtual elements[38]. This idea is not just an assumption; in fact, it can be deduced from the previously mentioned DT definitions[28,30,35]. Shao et al.[39] argue that although concepts behind DT might be old and known by simulation experts, DT is however a prominent step over the simulation model because classic simulations typically represent what happened in the past or may happen in the future based on initial assumptions, while DT focuses on what is happening right now and may be used to predict future states as well. VanDerHorn & Mahadevan[40] have the same viewpoint and argue that one reason for the potential confusion between simulation models and DT arises from the fact that while a simulation model is not necessarily a DT, the use of a simulation model combined with DT is prevalent.

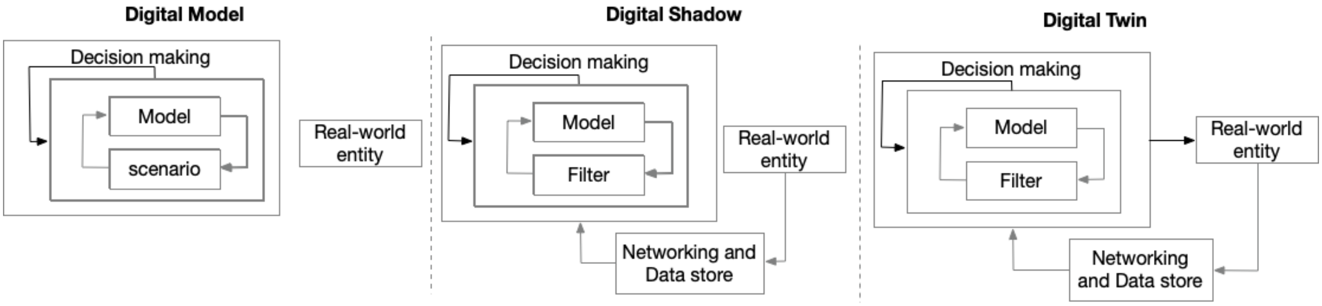

Figure 3. From digital model to DT according to Kritzinger et al.'s classification

Based on an evolved workflow, we look to clarify the evolution that has occurred.

3.1 M&S Workflow

As M&S is such a mature discipline, workflows exist for practitioners to guide them through a simulation study. The different concepts of the workflow are combined in the work of Balci, where a life-cycle model for M&S is defined[41].

Figure 4 defines a simplified view of such a workflow of a simulation study based on[41,42]. Activities are shown using ellipses. The control flow between different activities is shown with a full arrow. The dashed lines show the interaction with the system under study. We distinguish the following activities in the workflow:

1.1 Model Objective Definition: This activity defines the reasons for the simulation study. The problem of interest is defined. From this high-level question, the simulation analysts and domain experts define the specific questions of the study. These specific questions are translated into the properties of interest. Furthermore, the scope of the model is defined.

1.2 Create the Conceptual Model: The conceptual model is the model that is formulated in the head of the developer[41]. Zeigler defined a hierarchy of system specification that can be used as a foundation for creating a conceptual model[43].

1.3 Create a Programmed Model: The conceptual model must be captured in an executable/programmed model. Different programming languages and simulation formalisms are available to create the executable model. Once the programmed model is available, we can check if it is a good implementation of the conceptual model and does not contain any errors. This process is often referred to as *verification*.

1.4 Calibration or Parameter Estimation: The conceptual and programmed model are typically parametrized, so for a virtual experiment, these parameters must be assigned a value. Some parameters can be taken from component data sheets or literature. However, sometimes experiments need to be set up to measure the parameter. Finally, some parameters can only be estimated using optimization techniques to ensure the model's output is calibrated to the system's output.

1.5 Model Validation: Once a calibrated model is available, it still needs to be checked if it has any predictive capabilities within its domain of applicability. The process is called *validation*: "A computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model"[44]. Different techniques and statistical metrics are available for doing model validation[45].

1.6 Model Experimentation and Decision-making: The model can now be used for its purpose. In silico experiments are conducted using designed experiments. The results are used for decision-making, understanding a system, etc.

Note that most of these activities are done iteratively. Furthermore, the simplified life-cycle model does not show feedback loops to return to previous phases when needed. We refer to Balci[41] for a more detailed treatise of the life-cycle of M&S.

3.2 Evolution Towards DT Workflow

A simplified DT workflow is shown in Figure 5. This workflow is based on our experience of building digital twins and on typical model-based systems engineering workflows and standard software life-cycle models such as DevOps[46]. Compared to the simplified simulation study workflow, a lot has changed. A single step within this workflow contains the entire simulation study workflow. We see the following activities.

2.1 DT Objective definition: A digital twin is developed for a specific purpose, e.g., optimization of performance parameters of the system, control-oriented applications, monitoring and dashboarding. Based on these specific objectives of the DT, the developers create a set of requirements for the DT. The requirements and specifications also include the operational domain of the DT. The requirements translate into the specific properties of interest the DT needs to work on.

2.2 Model development life-cycle: This activity contains the life cycle shown in Figure 4. Based on the requirements and specifications of the DT, a model needs to be created or possibly reused from a library of models. M&S experts use the requirements of the DT to translate them into model requirements.

2.3 DT Architecture: The architecture of the digital twin is created. Decisions, such as using a distributed or

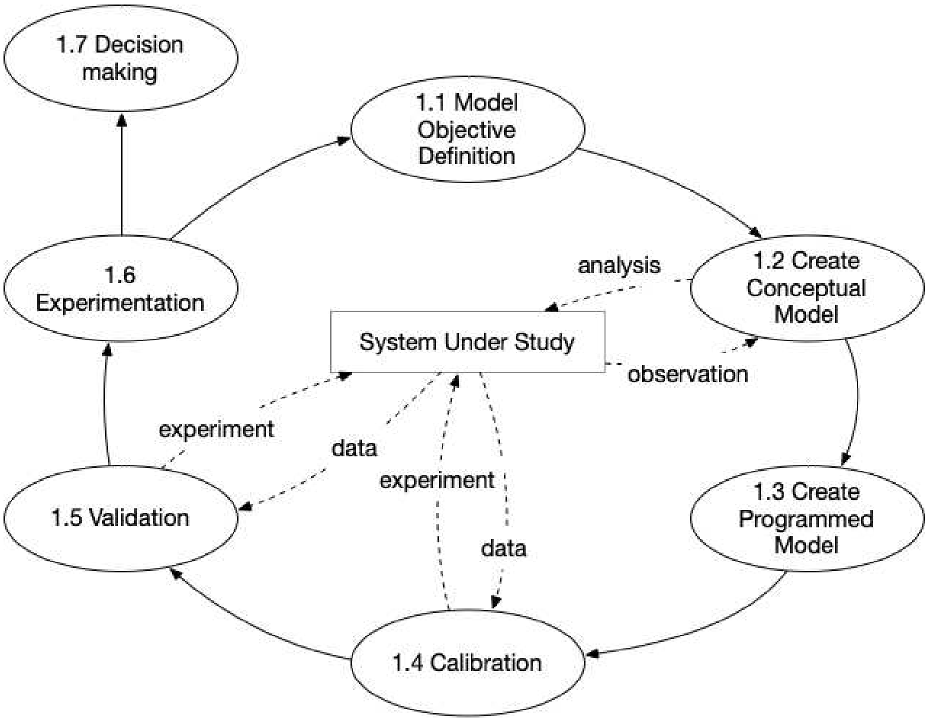

Figure 4. Simplified M&S workflow

centralized architecture, are taken. Besides the model, the DT needs many other components to operate, e.g., instrumentation, data collection, networking, data storage, and decision-making. Furthermore, the model is used for a specific purpose within the DT, e.g., what-if analysis or optimization. Decision-making components, based on the simulation outcomes, are defined.

2.4 DT Create/Build: Based on the architecture, the digital twin is developed, e.g., coding and testing of software components and setting up the networking and data infrastructure.

2.5 DT Deployment: Deployment is releasing the digital twin for use. The different data streams are connected to the DT. Deployment of a DT is typically done in cloud environments. However, fog and edge computing is also considered[47].

2.6 DT Verification and Validation: As a digital twin is a complicated software-based system, current software engineering practices should be considered. Verification is the process of checking if the services offered by the DT are created correctly. Validation checks that the services provided by the DT meet the needs of the system users. Note that verification starts in the DT build phase when testing the various components.

2.7 Data Collection: From the instrumented environment and system, the data is gathered by the system for use by the DT services and validation processes. Depending on the services, the data is also made persistent for later use.

2.8 DT Services: These are the services that implement the objectives of the digital twin. The services use the data and actuate the system. The service typically runs automated simulation experiments to support the automated decision-making provided by the DT service. Matta and Lugaresi classify these services as descriptive (e.g., health-monitoring), predictive (e.g., prognosis), and prescriptive (e.g., optimization) services[48].

2.9 DT Synchronization: The current state of the system must be estimated to be able to use the digital twin properly. These estimates can be used in the underlying models to initialize the models. Once the state is estimated, we need to be sure that the model that is used within the digital twin is up-to-date with the system and its environment. This is necessary because the system and environment can evolve over time (e.g., wear and tear, replacement of components). Furthermore, once it is detected that the underlying models are no longer valid, the model should be brought back in synchronization with the actual DT. The parameters must be updated if the model is valid for a larger operational domain (and/or a new initial state should be estimated). However, in some instances, a new model should be created or selected for use.

Note that the activities shown in grey, in Figure 5, are offline or development activities, while those in white are online or run-time activities. We also note that the semantics of this model might be slightly different compared to the semantics of Figure 4. While the digital twin is operational (and thus providing services), the continual validation techniques can run in parallel. Furthermore, a new model must be created when the current model is no longer valid while certain DT services are still up and running.

The two workflow models show that the DT workflow builds on top of the M&S workflow. From that perspective, DT can be seen as an evolution of M&S. However, when examined thoroughly, such evolution also brings new requirements for the concept to be truly feasible, i.e., to allow the simulation model to co-evolve with the actual system, capturing and reflecting its modifications. Defining new methods, techniques, and tools to address those requirements could lead to a revolution in the area of M&S.

4 Digital Twin: A Revolution?

The adoption of DT technology brings drastic changes both in the engineering and the practice of simulation methods and infrastructures. For example, at the engineering level, the usability of DT is most pronounced when a real system or object undergoes modifications over time, making the initial model of the object obsolete[35]. While in traditional M&S a new model has to be built, the DT model rather captures and reflects these modifications. At the practical level, DT simulation experiments are not based on assumptions on the initial conditions like in traditional M&S, but on current information available from the system[49]. Consequently, the space of possible initial conditions to explore is larger with a traditional M&S model than with the DT model.

In this section, we first revisit major requirements for DT engineering. Then we discuss some of the disruptive challenges that these requirements bring in the M&S field.

4.1 Requirements for Digital Twin Engineering

To create a successful DT, it must fulfill a set of requirements. These include both reinforced forms of M&S requirements and novel DT context-specific requirements. In what follows, we discuss the most prevalent requirements mentioned in the literature. These requirements typically are interdependent, and each one can include one or several sub-requirements.

4.1.1 Data value chain covers the critical need for the DT to access appropriate data pertaining to its real-world counterpart at the appropriate moment. Digital models collect and analyze huge amounts of data throughout the entire life cycle[38]. There is a need for a well-defined data hierarchy because, at each abstraction level DT provides data. Thus, the determination of handy information and data with considerable accuracy is vital[50]. Oliver identified an issue; for a DT there is still a need for adequate and reliable data sources and this problem has been the same for decades[39]. The identifiable information sources must retain information history and must be trustworthy, valuable, optimized, and available at any time for evaluating deterministic behavior, audit, and analytical purposes[51].

The International Telecommunication Union proposed a three-layered architecture of a digital twin network with a primary focus on three key subsystems, (i) unified data repository, (ii) unified data models, and (iii) digital twin entity management. The ITU also set out the requirements of a unified data repository as to be trustful and fast, it should provide a variety of data timely and accurately, be able to exchange real-time data within acceptable time delay, be easy to maintain, and be available at all times[52]. Reliability, integrity, and speed of data are crucial for system performance and for representing the state of a physical entity in real-time with an acceptable time delay because a system can be described based on its logic and input data[53].

Data value chain is the foundation of an end-to-end process responsible for collecting, processing, and supplying data from various sources, that could better be utilized for decision-making and optimization. Interoperability facilitates effective and efficient integration of the data across the different components of a DT. The next section highlights the interoperability requirements for a DT.

4.1.2 Interoperability is the ability of two or more DTs to exchange information and mutually use the information that has been exchanged[54]. Interoperability allows DT to communicate and exchange data between different DTs, simulation models, software, and platforms such that the same data can be used for different purposes, which can be achieved by establishing equivalence between various model representations[55,56].

Platform Interoperability is a DT's extension by using value-added services, such as AI, simulation, visualization, etc.[57]. However, *System Interoperability* is communication and interaction between DTs of different physical entities[57].

In a practical approach, few DT elements may already exist. In this case, there is no need to develop them as part of the new DT, but a need for a suitable interface to integrate the already existing DTs[50]. Integrating multiple simulations could be challenging. Open architectures and relevant interoperability standards can help to integrate different simulations at different fidelity levels[58]. The Digital Twin Consortium presented a complete DT interoperability framework based on seven key components: the system-centric approach, model-based interactions, holistic flow of information, state-based synchronization, heterogeneously distributed federated repository integration, actionable information exchange, and scalable mechanisms to streamline connectivity and collaboration in DT ecosystems[51]. Interoperability standards such as ISO 23247-4[36] and IEEE 1516[59] can be used for the mutual interaction and integration of heterogeneous DTs. Table 1 shows the review of different literature and their main focus in the area of DT interoperability.

Interoperability set out the foundations for effective collaboration and communication among the different DT components. However, this communication and collaboration need to be synchronized at a certain frequency and fidelity as defined by the Digital Twin consortium[32]. The next section addresses the synchronization aspects of DT and the subsequent section explores the fidelity requirements of DT.

4.1.3 Synchronization means that the state of the real system and its DT are kept consistent and up-to-date with each other's state using appropriate event-based or time-based methods[36,62]. Depending on the data flow in the DT, synchronization can be seen in both directions, from virtual to physical and physical to virtual[62]. In the former case, the synchronization concerns the tracking of the physical world by the virtual one, in the latter it concerns the control enacted by the virtual world upon the physical one. The level of synchronization (e.g., rate, quality, and volume) can vary and depends on the intended purpose[63]. Table 2 shows an overview of relevant literature and which synchronization

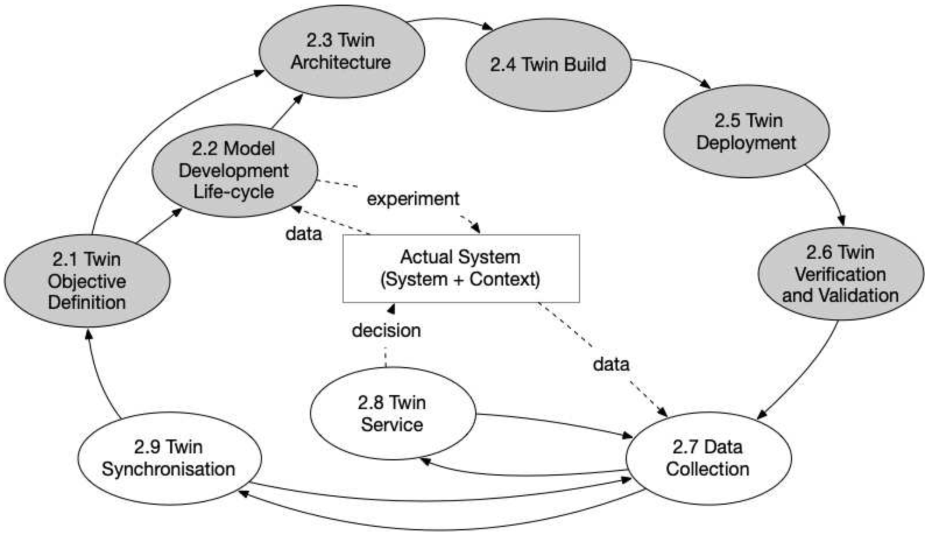

Figure 5. Simplified workflow for Digital Twins

Ref.	System Interop-erability	Platform Interop-erability	Integration
D.T.C.[51]	x	x	x
Lehner et al.[57]	x	x	x
Schleich et al.[55]	x		x
Durao et al.[56]	x		x
Wagner et al.[50]	x		
Shao et al.[58]	x		
Rasheed et al.[60]	x	x	x
Niaki et al.[61]		x	x

Table 1. DT interoperability literature considerations

Ref.	Physical to Virtual Sync.	Virtual to Physi-cal	Data Rate	Data Vol-ume
ITU[52]	x	x	x	
Jones et al.[62]	x	x	x	
Liu et al.[64]	x	x	x	x
Moyne et al.[63]	x	x	x	
Rasheed et al.[60]	x	x	x	x
Sjarov et al.[65]	x	x		
Meng et al.[4]	x	x	x	x
Talkhestani et al.[66]	x			
Wagner et al.[50]	x	x		
Wang et al.[67]	x			
Zhang et al.[68]	x	x		

Table 2. Synchronization considerations.

aspects are mentioned in their reviews/research. Most of the literature considers the bidirectional dataflows, such as in Grieves' definition, but fewer also consider the data rate and data volume considerations and their effect on the synchronization.

Synchronization ensures that the real-time representation of the virtual entity is closely aligned with the physical entity, while fidelity ensures the accurate representation of the complexities of the physical entity. The following section highlights the DT fidelity requirements.

4.1.4 Fidelity is vital for a DT to represent the current state of its physical counterpart as accurately as possible[56]. So, it is the fidelity of DT that determines nearness to the real counterpart.

From the perspective of M&S, it can be argued that DT needs high-fidelity simulation modeling technology, where this fidelity is not only related to the fidelity of the model construction but also to the data-related issues, e.g., accuracy and frequency[68]. While certain scholars discuss fully mirroring[66], ultra-realistic[65] and ultra-high-fidelity simulation[69,70], others seek to establish fidelity levels to optimize the DT's advantage to existing challenges[36,40,71]. Academic definitions commonly imply that a high-fidelity model is a crucial requirement for a DT. However, practical use does not always require high fidelity, as it can be costlier. Therefore, it is crucial to determine the appropriate level of fidelity for DT[72]. Jones et al, 2020[62] argue that DT fidelity levels comprise multiple dimensions, including the number of parameters, the precision of those parameters, and the degree of abstraction in the reciprocal exchange between the virtual and physical twins. The appropriate fidelity level is not necessarily the highest level of model fidelity feasible[40], and is dependent on the use-case. It seems that While DT fidelity pertains specifically to the suitable precision of the model's representation, DT validity encompasses the broader notion of whether the DT serves as a fitting and efficient tool for its intended application. The subsequent section discusses the Verification and validation aspects of DT.

4.1.5 Verification and validation (V&V) of DTs pertain to the assurance of constructing a DT in alignment with its objectives (validation) and ensuring its accurate implementation (verification)[73]. DTs typically include different models, components, sub-components, and processes, which necessitates V&V on an individual basis as well as for the entire system[40,55]. To maintain the model's validity throughout the entire life cycle; clear and well-defined guidelines are needed for V&V[58]. A DT must have a validated specification of what to simulate and what to predict, with which input, and which approach[50]. The system's output should be continuously compared and monitored with a reference point to detect errors and anomalies[60].

Primarily DT concept applies to highly automated systems. When there is human involvement and decision-making, then due to the high degree of randomness of human actions, the system cannot be perfectly shadowed virtually[39], thus directly affecting the V&V of DT.

Hua et al.[74] argue that we may need a two-layer approach for V&V of DT, one at the system level and the other at the constituent system level. In general, due to the dynamic nature of DTs, V&V should be a continuous process (either online or offline) that needs to be performed periodically (or on-demand)[38]. Sargent stated that V&V of a basic simulation model comprises four pillars; (i) Conceptual Model Verification, (ii) Computerized Model Validation, (iii) Operational Validation, and (iv) Data Validation[75]. Lugaresi et al.[38] presented the DT V&V approach based on the following four levels; (i) Logic-level Validation (Digital Model), (ii) Input-level Validation (Input Data), (iii) Event-level validation (System Events) and (iv) Performance-level Validation (KPIs). Table 3 shows the comparison between different literature and their considerations for the V&V of DT.

For a well-established and accurate system, there could be a need for system improvement, reduction, or extension depending on the requirements. These modifications could be at the core architectural level of the system, or either at the output or input level of the system. These are categorized as extensibility and scalability; the successive sections address these issues.

Ref.	Online DT V&V	Offline DT V&V	Data V&V	General System V&V
ITU[52]			x	
Lehner et al.[57]			x	x
Sargent[75]				x
Lugaresi et al.[38]	x			
Khan et al.[76]	x	x		x
Peter et al.[39]				x
Dahmen et al.[77]				x
Hua et al.[74]	x	x		x
Shao et al.[58]				x
Wagner et al.[50]			x	x
Locklin et al.[?]	x	x	x	x
Rasheed et al.[60]			x	
Schleich et al.[55]				x

Table 3. DT V&V literature considerations.

4.1.6 Extensibility refers to the DT's capability to integrate, add, or replace models[55,56], that allows a DT to expand or enhance easily. Extensibility allows DT to accommodate new applications and functionalities without significant effort. The evolution of DT must be aligned with its physical counterpart[57], while maintaining its backward compatibility[52,60]. The DT functionalities should smoothly extend their capabilities with no effect on existing functions[52,63]. Extensibility requirements can vary and may encompass subordinate requirements, notably modularity and standardization.

Extensibility can vary from small sensor integration to a whole new model integration within the already existing DT ecosystem. Thus, it is an important aspect to focus on at the time of designing a DT architecture. It could directly affect the cost and other related aspects. When a system evolves over time, it has a direct effect on its parameters and data, which gives rise to the problems of scalability. The DT scalability is described in the following section.

4.1.7 Scalability refers to a DT's capability to show the state of its real counterpart at different dimensional, temporal and spatial scales (microscopic scale - fine detail, mesoscopic scale - medium detail, and macroscopic scale)[55,56]. Processing data at different levels of granularity contributes a lot to the holistic understanding of the modeled entity. Multiscale simulation has been recognized as one of the most important visions of DTs[64]. To enhance the scalability, models should support different dimensions, spatial, and time scales[68]. DT must be capable of automatically adjusting the scale of the virtual twin regarding the growth or shrink of its physical entity[52]. Like extensibility, scalability could also be sometimes a constraint and could limit the reduction, modification, or extension of a system to a step above or below. Some systems change constantly, few grow rapidly, and others evolve over time, thus, they have fuzzy borders[78].

In addition to the aforementioned requirements, DT needs to be explainable to the user. Explainability enhances user trust. It helps to describe a system and how it processes data and makes conclusion and predictions.

4.1.8 Explainability is an ability that aims to provide insight into how a DT can be understood by the user entity. According to ISO 23247[79], the user can be human, applications, or other systems that use the DT. When a human being is a DT user, visual representation is essential to provide them with comprehensible outputs[80]. The International Telecommunication Union (ITU) emphasizes that all the elements of DT, such as data and models, should be developed by means of visualization to provide better access for involved humans[52]. Commonly, the M&S community focuses more on model explainability, which facilitates the collaboration and interaction between models and users to provide a consistent understanding[67]. Table 4 provides a brief overview of the relevant literature, highlighting the explainability considerations in terms of interaction [A], comprehensiveness [B], semantics [C], intelligence [D], and abnormal data [E] discussed in their respective studies.

4.2 Disruptive challenges

The requirements for DT engineering previously presented bring with them a set of open research challenges. We discuss those that from an M&S perspective appear as potentially disruptive.

4.2.1 Dynamic State Estimation

State estimation is the challenge of determining the actual state of the system in operation. *Dynamic state estimation*

Ref.	A	B	C	D	E
Zhu et al.[80]	x	x		x	
Shao et al.[39]		x			
ITU[52]	x				
Zhang et al.[68]	x			x	x
Sjarov[65]					
Wang[67]		x			
Wagner et al.[50]		x	x		
Rasheed et al.[60]	x				
Sjarov et al.[65]		x	x	x	

Table 4. Explainability Considerations.

refers to the process of determining how a system is operating when its state is changing dynamically based on real-time observation data. Dynamic state estimation is required because, in most circumstances, noisy and incomplete observation data from dynamic systems make it impossible to derive the system state directly from the observation data.

One of the key features of the DT is the integration of real-time data with a digital model to support real-time prediction/analysis of the system under study (similarly to RtS[14,18–21,81]). Most of the DTs model the dynamic behavior of the corresponding physical systems. Therefore, they need the dynamic state estimation.

To enable simulation-based real-time prediction/analysis, Hu proposes a framework of data assimilation for dynamic systems in operation[82]. The goal of the framework is to support real-time decision-making for the system under study. To accomplish this, simulation-based future behavior prediction and analysis of the system is required. The simulation-based prediction and analysis depends on a precise evaluation of the system under study's current state in real-time, which asks for dynamic state estimation, therefore a simulation-based prediction can be used. Moreover, to accurately characterize the system in operation through simulation-based prediction and analysis, the simulation model becomes essential. This requirement calls for online model calibration of the model parameters based on real-time data gathered from the system. The data assimilation approach addresses both dynamic state estimation and online model calibration activities by merging information from real-time data and the simulation model.

Data assimilation has been recently used for discrete-event and discrete-time systems, including agent-based models. The particle filter (PF) approach is frequently a viable choice for stochastic simulation models of discrete systems due to its non-linearity and non-Gaussianity[83]. But it is computationally expensive because of the probability distributions of model runs. In their findings, they observe that the choice of time intervals, rather than the number of particles, more strongly influences the estimation accuracy of such a system utilizing PF. When measurement errors are underestimated, state estimates are poorer than when measurement errors are overestimated. Better state estimations are not a given just because one has a proper understanding of the measurement errors. In addition, over-estimation of errors yields better state estimation and is more sensitive to rapid system changes.

4.2.2 Online and Continual Validation Continual validation is the process of continually ensuring a DT, or more concretely the model(s) in the DT, remains a valid representation of their real-world counterpart. When the model becomes invalid, e.g., due to changes in the real-world system, a recalibration is needed to match the DT to the real world once again, as was shown in stage 2.9 in Figure 5. This is conceptually different from model validation in M&S where, typically, a calibration attempt is performed first, after which the model is subjected to the validation procedure. Once positively validated, the model is generally also assumed "finished". Besides this conceptual difference, there is a high level of similarity between traditional model validation and the validation of models in DTs. As such, the model validation techniques from M&S can largely be carried over[84]. However, one faces several challenges when continually attempting to apply those model validation techniques. They stem from the fact that only the runtime data of the system in operation is available. This leads to the following challenges:

- In traditional model validation, a validation experiment is carefully defined. With DTs however, data is streamed continuously from the physical system. Therefore, one must define which part of this data stream can be considered an experiment for validation. Stated differently, you need to delineate the experiment in the data stream.

- In traditional model validation, experiment replications are controlled by the experimenter. With DT data, we are relegated to grouping or batching data from equal "experiments" (which have been delineated as stated previously). In this batching procedure, it is important to choose a proper time horizon, e.g., when using data from the last month, there is a risk of averaging out any of the changes that we would want to observe and check against.

- In a traditional validation experiment, the bounds/ranges are carefully controlled. With a DT, we are limited to the bounds that occur naturally from the system's routine/regular operation. A problem with these bounds is that they are usually only a subset of the entire range for which the utilized simulation models were validated at design time. As such, the range you can continuously validate against is limited. A potential workaround for this problem is that of "experimental runs" where we instruct the system to perform an experimental execution, which must still achieve the regular goal but in such a way that it yields additional information content[85].

Continually validating a model has its benefits; the model can be continually checked for correctness. However, the available data is not as information-rich as those gained from a specifically crafted model validation experiment. For physics-based models, traditional literature on this topic therefore ought to be reviewed[42,45,86].

In the case of DTs of production facilities/smart manufacturing, the used models are often discrete-event queuing models, with arrival times and processing times characterized by stochastic distributions. In such cases, perhaps new validation techniques are needed. In literature, different metrics are calculated on periods in the data streams, combined with thresholding to trigger model updating[38,87,88]. The use of stochastic simulation also adds another caveat, which is that both the arrival/input distribution and the logic/model itself could become invalid, and a good validation ought to be able to pinpoint exactly which of these two has changed[87].

This idea of pinpointing errors at runtime leads us to another mature field where methods could be found to aid in this challenge of continual validation: the field of fault detection and diagnosis. Recall that the goal is to detect divergence between the model in the DT and the twinned system, with a recalibration in case of divergence. This is conceptually not that different from fault detection and diagnosis, where the goal is to detect faults in a physical system, diagnose/isolate them and take corrective action based on their identification[89], the main difference being that with the DT, we assume the real-world system is the ground truth and any faults occurring in it should propagate back to the digital model. Not all techniques from this field carry over, e.g., physical redundancy techniques cannot be applied to DTs. Still, model-free techniques such as trace inspection with limit checking[90] or model-based techniques do carry over. Model-free techniques operate on the traces of the model and combine the filtering of those traces with limit checking to detect faults, similar to Lugaresi et al.'s work[87]. Model-based methods use a digital model that produces traces in parallel to the physical system, and through trace comparison, faults can be detected.

Furthermore, these techniques find their application for continuous[89], hybrid[91] and discrete-event[92,93] systems. Normally, these techniques were also designed to operate at system runtime; as such, they are generally not particularly heavy on the computational side, making them useable for real-time applications.

The previously discussed validation techniques rely on the transmission of data from the physical twin to the virtual space. It's therefore also paramount that this data is flawless. With this in mind, we can state that not only should we validate the models in the digital twin, we ought to also validate the data as it arrives at the digital twin. The initial culprit to blame for faulty data would be the sensors on the physical system that collect the data. Therefore, any digital twin system could benefit from sensor fault detection and isolation[94–96]. In fault detection and isolation, the sensor faults are generally classified as incipient or abrupt failures. In an incipient failure, the sensor is working in an abnormal or deteriorated way. In an abrupt failure, the sensor suddenly stops working. Various diagnosis methods can be applied to detect and isolate the fault, such as model-based, knowledge-based, and deep learning based approaches. In[96], a set of machine learning techniques are applied for the sensor fault detection in a digital twin specifically. While a likely culprit for faults, the sensor is not the only place where faults can be introduced, so is the communication network, and any layers of software that perform processing/packing/unpacking of data[95]. As such, perhaps the data validation should be performed right before it is fed into the digital twin. We see this idea applied in the field of machine learning, where data is aggregated from multiple sources before being fed into a machine learner[97]. These techniques should be integrated into the digital twin's data pipeline, for the digital twin to work optimally, but also for the continual validation to work correctly.

In summary, more work is needed to get to continual validation, but because of the similarity to stage 1.5 in the M&S workflow of Figure 4, there exist techniques that can help us along. We also make one note regarding the use of continual over the generally accepted continuous (such as in continuous testing, integration, and deployment). Nowadays, continuous has the implication that it is an everlasting process, whereas continual implies some periodicity in the process but with pauses. We think that this is a more correct representation of how the process is to be implemented, which is why we opt for continual.

4.2.3 Automated Recalibration and Co-evolution

When a model no longer accurately represents the behavior of the real-world system it models, changes must be made to that model. Two scenarios are possible: (a) either the parameters of the model no longer accurately reflect the system and its environment, in which case a parameter calibration within the model's range of validity suffices. (b) the system is no longer within the valid context of the model, in which case a new model must be selected/developed that again fits the system's current context. In either case, reinitialization of the model is necessary. In[98], this problem is described along with an accompanying workflow, but no solution is given for the reinitialization step that brings the model back in synchronization with the real system.

One way to deal with the second scenario is to use meta-information about the validity of a model in a certain context. In literature, Zeigler[43] defined the concept of the experimental frame as *"The conditions under which the system is observed and experimented with"*. The idea of the experimental frame is to make the contextual information about the simulation model explicit. In doing so, it gains a dual purpose: it implements meta-data that is needed to specify the range of validity, and it defines an operational view of the experiment using a generator, transducer, and acceptor. The concept was further refined by Traore and Muzy[99]. The experimental frame could be used to check if the context of the model is still valid. Denil et al.[100] looked at the uses of such an experimental frame: checking for a new context, calibration, searching for a model in a library of simulation models, reproducibility, etc., and concluded that it might not be defined well enough for these purposes. In the context of DTs, no processes are defined to allow for automated calibration experiments. Validity frames[100–102] evolve the concepts defined in the experimental frame (the experimental frame is embedded within). It has the meta-data needed to reason over the model and run simulations (such as initial conditions, parameter ranges, model architecture and rationale, etc.) and the operational view where signal monitors are generated to check the model's bounds at run-time. It also adds workflows for different activities within the M&S process, such as calibration and validation. This could be used as a starting point for checking if the model could be recalibrated, if a new model should be selected from the model library, or if a new model should be developed.

Similar methods and tools have been created in the co-simulation community that can be used as a starting point: Otter et al.[103] propose to annotate the parameters of a Modelica model with traceability, uncertainty, and calibration information to improve model quality, thus increasing correct use of models. Instead of relying on external data formats, they insert this machine-readable metadata within the models. The Modelica association also developed a standard for creating co-simulation packages: System Structure and Parameterisation (SSP)[104]. The SSP could be extended to allow for structure verification, parameter verification and boundary adequacy testing.

Another source of inspiration is the control community. For example, the MAPE-K loop which is a high-level feedback control loop from IBM[105] for self-adaptive systems, has been integrated in DTs[106]. This approach is based on the principle of changing the DT model when an anomaly is detected. The MAPE-K architecture makes a distinction between the domain-specific system, the managed system, and the system manager. The managing system contains four phases that use common knowledge to: (a) monitor the managing system and its context, (b) analyze the situation and decide if adaptations are required, (c) plan the adaption to this new configuration and (d) execute the transition to this new configuration using a mode-changing protocol. In this situation, the change of model happens when an anomaly is detected; the new model requires calibration and the controller needs re-optimization.

However, automatically creating a new model seems to be a difficult problem. One way to deal with this is by searching for alternative models based on the current model. David et al. use reinforcement learning techniques for this specific purpose[107]. Another approach is that of the control community, which has long worked on the system identification problem. In system identification, statistical methods are used to create a black-box model of the system[108]. These techniques can be integrated into the DT if a new white box model needs to be created.

4.2.4 Real-time adaptive operations

Real-time aspects pertain to the fact that there may be some factor of timeliness required from the data used in the DT. The stringency of this requirement depends on the goals of the DT. When the DT acts in some form of process control, soft or hard real-time constraints may be required. Otherwise, the non-strict, human, notion of real-time suffices[109].

Some real-time aspects require investigation in the DT concept:

The availability of quantitative data and advanced analytics in real-time via DT enables better-informed and faster decision-making. In particular situations, decision-making processes must adhere to real-time constraints. As a result, the DT model should be sufficiently fast to make decisions within the specified timescale while also accurate enough[35]. The computational cost of using a more complex model is often excessively expensive, and the system might fail to meet the deadline. There are several approaches to dealing with computationally expensive models:

- Multi-resolution modeling (MRM) is the process of creating a single model, a family of models, or both to represent the same phenomenon at several resolution levels while allowing users to enter parameters at each level according to their needs.[110]. MRM is also known as variable- or selectable-resolution modeling. Sometimes the word fidelity is used instead of resolution. MRM is closely related to model abstraction, which is a way of simplifying models while keeping the essence of a phenomenon concerning the application at hand[111].

- Franceschini et al.[112] present an adaptive abstraction approach. They utilize a specified trigger to determine when to transition between abstraction levels. A dynamic abstraction simulation that alternates between an agent-based formalism and a discrete event formalism is presented by the author in prior work. The statistical analysis of the observed emergent behavior serves as the basis for the decision to change abstraction levels. A more rigorous framework[113] extends the adaptive abstraction technique to decide when and where to switch between abstraction levels.

- Some research recommends employing an abstracted and/or approximated model instead of a more detailed model. The *self-Adaptive Abstraction and Approximation* technique[114] is based on the MAPE-K loop previously presented to adapt a real-time system under study by changing the model and using an approximated and/or abstracted model instead of the more detailed model. However, the validation of the substitute model is an essential issue that needs more investigation. One approach is to look at the model behavior, calculate the deviation, and find tolerances[115,116]. Another approach is using the ESS (EMF-Based Simulation Specification) technique[117].

Real-time communication is another challenging part of the real-time aspect. The communication rate between the real world and the system is something to consider, as it is not feasible to communicate at every microsecond.

4.2.5 Sustainability

This relates to the observation that DTs use large amounts of computational resources to provide their different services. Computing as an industry is currently responsible for 2%[118] to 6%[119] of the emissions of greenhouse gasses globally, with a predicted share of 6%[120] (22%[119]) in 2040 (2030), therefore we must require future DTs to be sustainable in regard to energy consumption. Reasoning over energy and power consumption and their associated models can include several levels of impact[121]:

- First order impacts: Impact via the design and operation of the DT.

- Second order impacts: Secondary impact related to the effect of DTs on, e.g., production and product usage. For example, the decrease of energy consumption of a device because of the optimization possible by the digital twin.

- Third order impacts: indirect effects caused by DTs, e.g., impacting an industry's structure or the lifestyle of persons.

To create sustainable DTs, all the above aspects should be considered. Bellis et al. propose an additive model where the consumption of the energy occurs [122]:
$E_{total} = E_{design} + E_{local} + E_{network} + E_{cloud} + E_{update}$
where:

- E_{design} is the energy consumed for creating the digital twin. Building a simulation model of a digital twin might not greatly impact this factor. However, this term might have a significant impact when using data-driven methods.
- E_{local} is the energy consumption at the analogue side of the system (e.g., by storing the data, pre-processing the data, and executing a part of the digital twin model locally).
- $E_{network}$ is the system's energy consumption by sending and receiving messages on the network.
- E_{cloud} is the energy consumption by executing the digital twin in the cloud environment.
- E_{update} is the energy necessary to redesign and update the model during the system's life cycle.

For the construction of DTs, each of the different sources of energy loss should be further examined. Modeling and simulating the full infrastructure and the design process seems a logical first step.

5 Conclusion and Perspectives

The Digital Twin concept has surfaced with the prominence of data and virtual technologies for the analysis, design, and control of smart systems in the ever-growing context of "smart everything", from industrial and health sectors to educational and urbanization sectors. However, as new fields usually take time to coalesce to form generally accepted definitions, the concept of DT is differently approached and defined by different professional communities.

In this paper, we approach the DT concept from the viewpoint of M&S practitioners. In some ways, we see the DT concept as a natural extension of M&S practices, and we give an overview of how this evolution happened. We also see the apparent divergence of understandings of the DT concept as the potential that there is something revolutionary to it in M&S. We discuss some of the disruptive challenges to be addressed in that context.

Numerous research efforts are being conducted in the M&S field towards more developments of the DT technology. Among them, we retain the following ones, for they seem to draw innovative (and potentially disruptive) perspectives for the next years:

- Paredis and Vangheluwe [123] introduced the concept of *Digital Z*. A novel method of recognizing twinning frameworks is identified by the term, which refers to the demographic generation naming convention such as Generation X, Generation Z, Generation Alpha, etc. Digital Z is defined where Z can be a model, shadow, twin, passport, avatar, etc. Systems engineering uses multiple Digital Zs, frequently in combination, for many different kinds of goals. For this reason, a variety of architectures are provided for various Digital Zs. The engineers' goals guide the construction of each Digital Z architecture. The idea is to use each Digital Z for each property of interest such as safety, average energy consumption and so on.
- Niyonkuru and Wainer [124] introduced the concept of *Digital Quadruplet* in order to improve the development of Embedded Real-Time Systems: a 3D virtual replica of the real world under study (which is called here the DT), a discrete-event formal model of the system of interest that can be used for formal analysis as well as simulation studies (called the Triplet), and a physical model of the real system under investigation for experimentation (called Quadruplet).
- Traoré and Ducq [125] introduced the concept of Digital Industrial Territories (DITs), to be foreseen as the next step in the on-going industrial revolution. The technological ambition of this approach is to realize an effective vision of Digital Enterprises (DTs of Enterprises) within Digital Supply Chains (DTs of supply chains). Indeed, the Information Technology environments within industrial companies, ranging from embedded systems on shop floor level to operations and manufacturing execution systems or resource planning systems, form a basis for the vision of a digital management of the production plants. Each profile is a digital enterprise with DTs that can be coupled with the DTs of other profiles, leading to the digital supply chain of the network of enterprises then created. In that way, geographically distributed enterprises can form larger DT-driven consortia, abolishing spatial constraints on the monitoring and control actions, and the overall management of operations.
- David and Syriani [107] proposed an approach for the automated construction of simulators based on the inference of DEVS (Discrete Event System Specification) models by reinforcement learning. The reinforcement learning agent has an action list to build a DEVS model and its reward is obtained by comparing the traces of the built DEVS model with the traces of the system. The agent is based on the trial and error approach to Markov decision.

When we started this discussion, the question raised by the title of the paper had two explicit alternatives (is DT an evolution or a revolution in M&S?), but also an implicit one (is it just another buzzword?). We leave it to the reader to answer the question based on the background provided in this paper.

Acknowledgements

Zeeshan Ali and Milad Poursoltan are funded by the ACT program at the University of Bordeaux (ANR-20-IDES-0001). Raheleh Biglari is funded by the BOF fund at the University of Antwerp. Joost Mertens is funded by the Research Foundation - Flanders (FWO) through strategic basic research grant 1SD3421N.

References

1. Goldsman D, Nance RE and Wilson JR. A brief history of simulation revisited. In *2010 Proceedings of the 2010 Winter Simulation Conference*. IEEE, pp. 567–574.
2. Zeigler BP, Nicolau de França BB, Graciano Neto VV et al. *History of Simulation*. Cham: Springer International Publishing. ISBN 978-3-031-11085-6, 2023. pp. 413–434. DOI:10.1007/978-3-031-11085-6_17. URL https://doi.org/10.1007/978-3-031-11085-6_17.
3. Oden T and Ghattas O. Computational science: The "third pillar" of science. In *2014 The Academy of Medicine, Engineering & Science of Texas'(TAMEST's) Annual Conference January*. pp. 16–17.
4. Meng W, Yang Y, Zang J et al. Dtuav: a novel cloud–based digital twin system for unmanned aerial vehicles. *Simulation* 2023; 99(1): 69–87.
5. Kritzinger W, Karner M, Traar G et al. Digital twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline* 2018; 51(11): 1016–1022.
6. Vallée A. Digital twin for healthcare systems. *Frontiers in Digital Health* 2023; 5: 1253050.
7. Bao J, Guo D, Li J et al. The modelling and operations for the digital twin in the context of manufacturing. *Enterprise Information Systems* 2019; 13(4): 534–556. DOI:10.1080/17517575.2018.1526324. URL https://doi.org/10.1080/17517575.2018.1526324. https://doi.org/10.1080/17517575.2018.1526324.
8. Farsi M, Daneshkhah A, Hosseinian-Far A et al. *Digital twin technologies and smart cities*. Springer, 2020.
9. Gartner, Inc. Hype cycle for emerging technologies 2017, 2017. URL https://www.gartner.com/en/documents/3768572.
10. Gartner, Inc. Hype cycle for emerging technologies 2018, 2018. URL https://www.gartner.com/en/documents/3885468.
11. Wortmann A. Digital twin definitions, 2023. URL https://awortmann.github.io/research/digital_twin_definitions/.
12. Allen BD. Digital twins and living models at NASA. In *2021 Digital Twin Summit*. URL https://ntrs.nasa.gov/citations/20210023699.
13. Roberts SD and Pegden D. The history of simulation modeling. In *2017 Winter Simulation Conference (WSC)*. IEEE, pp. 308–323.
14. Tarnawski J and Karla T. Real-time simulation in non real-time environment. In *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, pp. 577–582.
15. Rubinoff M. Digital computers for real-time simulation. *Journal of the ACM (JACM)* 1955; 2(3): 186–204.
16. Forrester JW and Everett RR. Project whirlwind collection. URL https://archivesspace.mit.edu/repositories/2/resources/1157.
17. Everett RR. The whirlwind i computer. In *Papers and discussions presented at the Dec. 10-12, 1951, joint AIEE-IRE computer conference: Review of electronic digital computers*. pp. 70–74.
18. Popovici K and Mosterman PJ. *Real-time simulation technologies: principles, methodologies, and applications*. CRC Press, 2017.
19. Lugaresi G and Matta A. Real-time simulation in manufacturing systems: Challenges and research directions. In *2018 Winter Simulation Conference (WSC)*. IEEE, pp. 3319–3330.
20. Rogers P and Gordon R. Simulation for real-time decision making in manufacturing systems. In *1993 Proceedings of 1993 Winter Simulation Conference - (WSC '93)*. pp. 866–874. DOI:10.1109/WSC.1993.718331.
21. Alison Harper NM and Pitt M. Increasing situation awareness in healthcare through real-time simulation. *Journal of the Operational Research Society* 2023; 74(11): 2339–2349. DOI:10.1080/01605682.2022.2147030. URL https://doi.org/10.1080/01605682.2022.2147030. https://doi.org/10.1080/01605682.2022.2147030.
22. Gelernter D. *Mirror worlds: Or the day software puts the universe in a shoebox... How it will happen and what it will mean*. Oxford University Press, 1993. URL https://www.google.fr/books/edition/Mirror_Worlds_Or_The_Day_Software_Puts_t/PmCoPwAACAAJ?hl=en.
23. Davis WJ. On-line simulation: Need and evolving research requirements. *Handbook of simulation* 1998; 465: 516.
24. Fujimoto R, Lunceford Jr WH, Page EH et al. Grand Challenges for Modelling and Simulation (Dagstuhl Seminar 02351). Dagstuhl Seminar Report 350, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2002. DOI:10.4230/DagSemRep.350. URL https://drops.dagstuhl.de/entities/document/10.4230/DagSemRep.350.
25. Aydt H, Turner SJ, Cai W et al. Symbiotic simulation systems: An extended definition motivated by symbiosis in biology. In *2008 22nd Workshop on Principles of Advanced and Distributed Simulation*. pp. 109–116. DOI:10.1109/PADS.2008.17.
26. Onggo BS. *Symbiotic Simulation System (S3) for Industry 4.0*. Cham: Springer International Publishing. ISBN 978-3-030-04137-3, 2019. pp. 153–165. DOI:10.1007/978-3-030-04137-3_10. URL https://doi.org/10.1007/978-3-030-04137-3_10.
27. Cao Y, Currie C, Onggo BS et al. Simulation optimization for a digital twin using a multi-fidelity framework. In *2021 Winter Simulation Conference (WSC)*. pp. 1–12. DOI:10.1109/WSC52266.2021.9715498.
28. Grieves M and Vickers J. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*. Cham: Springer International Publishing. ISBN 978-3-319-38756-7, 2017. pp. 85–113. DOI:10.1007/978-3-319-38756-7_4. URL https://doi.org/10.1007/978-3-319-38756-7_4.
29. Shafto M, Conroy M, Doyle R et al. Modeling, simulation, information technology & processing roadmap. *National Aeronautics and Space Administration* 2012; 32(2012): 1–38. URL https://www.nasa.gov/sites/default/files/501321main_TA11-ID_rev4_NRC-wTASR.pdf.
30. Glaessgen E and Stargel D. *The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles*. 2012. DOI:10.2514/6.2012-1818. URL https://arc.aiaa.org/doi/abs/10.2514/6.2012-1818. https://arc.aiaa.org/doi/pdf/10.2514/6.2012-1818.

31. Alliance Industrie du futur. Digital twin: Leveraging the digital transformation of the industry, 2023. URL http://www.industrie-dufutur.org/content/uploads/2023/05/AIF_JumeauNumeriqueEN_US.pdf.
32. Digital twin defination. *Digital Twin Consortium* ; URL https://www.digitaltwinconsortium.org/initiatives/the-definition-of-a-digital-twin/.
33. Lo C, Chen C and Zhong RY. A review of digital twin in product design and development. *Advanced Engineering Informatics* 2021; 48: 101297. URL https://doi.org/10.1016/j.aei.2021.101297.
34. General Electric. What is a digital twin?, 2023. URL https://www.ge.com/digital/blog/what-digital-twin.
35. Wright L and Davidson S. How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences* 2020; 7(1): 1–13.
36. ISO 23247-1:2021 Automation systems and integration-Digital twin framework for manufacturing - Part 1: Overview and general principles. Standard, International Organization for Standardization, Geneva, CH, 2021.
37. Babic BR. Digital twins in smart manufacturing. In *Soft Computing in Smart Manufacturing*. Walter de Gruyter GmbH, 2021.
38. Lugaresi G, Gangemi S, Gazzoni G et al. Online validation of digital twins for manufacturing systems. *Computers in Industry* 2023; 150: 103942. DOI: https://doi.org/10.1016/j.compind.2023.103942. URL https://www.sciencedirect.com/science/article/pii/S0166361523000921.
39. Shao G, Jain S, Laroque C et al. Digital twin for smart manufacturing: the simulation aspect. In *2019 Winter Simulation Conference (WSC)*. IEEE, pp. 2085–2098.
40. VanDerHorn E and Mahadevan S. Digital twin: Generalization, characterization and implementation. *Decision support systems* 2021; 145: 113524.
41. Balci O. A life cycle for modeling and simulation. *Simulation* 2012; 88(7): 870–883.
42. Law AM and Kelton D. *Simulation modeling and analysis*, volume 3. Mcgraw-hill New York, 2007.
43. Zeigler BP, Praehofer H and Kim TG. *Theory of modeling and simulation*. Academic press, 2000.
44. Schlesinger S, Crosbie RE, Gagné RE et al. Terminology for model credibility. *SIMULATION* 1979; 32(3): 103–104. DOI:10.1177/003754977903200304. URL https://doi.org/10.1177/003754977903200304. https://doi.org/10.1177/003754977903200304.
45. Oberkampf WL and Roy CJ. *Verification and Validation in Scientific Computing*. Cambridge: Cambridge University Press, 2010. DOI:10.1017/CBO9780511760396.
46. Kim G, Humble J, Debois P et al. *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. IT Revolution, 2021.
47. Pushpa J and Kalyani S. Using fog computing/edge computing to leverage digital twin. In *Advances in Computers*, volume 117. Elsevier, 2020. pp. 51–77.
48. Matta A and Lugaresi G. Digital twins: Features, models, and services. In *2023 Winter Simulation Conference*.
49. Rosen R, Von Wichert G, Lo G et al. About the importance of autonomy and digital twins for the future of manufacturing. *Ifac-papersonline* 2015; 48(3): 567–572.
50. Wagner S, Milde M, Barhebwa-Mushamuka F et al. Digital twin design in production. In *2022 Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems: Proceedings of the 8th Changeable, Agile, Reconfigurable and Virtual Production Conference (CARV2021) and the 10th World Mass Customization & Personalization Conference (MCPC2021), Aalborg, Denmark, October/November 2021 8*. Springer, pp. 339–346.
51. Budiardjo A and Migliori D. Digital twin system interoperability framework. *Digital Twin Consortium* 2021; .
52. ITU-TY.3090 - Digital twin network – Requirements and architecture - Y series: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities, Digital twin network – Requirements and architecture. Standard, International Telecommunication Union (ITU), 2022.
53. Nelson B et al. *Foundations and methods of stochastic simulation*. Springer, 2021.
54. nformation technology — Cloud computing — Part 1: Vocabulary. Standard, International Organization for Standardization, Geneva, CH, 2023.
55. Schleich B, Anwer N, Mathieu L et al. Shaping the digital twin for design and production engineering. *CIRP annals* 2017; 66(1): 141–144.
56. Durão LFC, Haag S, Anderl R et al. Digital twin requirements in the context of industry 4.0. In *2018 Product Lifecycle Management to Support Industry 4.0: 15th IFIP WG 5.1 International Conference, PLM 2018, Turin, Italy, July 2-4, 2018, Proceedings 15*. Springer, pp. 204–214.
57. Lehner D, Pfeiffer J, Tinsel EF et al. Digital twin platforms: requirements, capabilities, and future prospects. *IEEE Software* 2021; 39(2): 53–61.
58. Shao G and Kibira D. Digital manufacturing: Requirements and challenges for implementing digital surrogates. In *2018 Winter Simulation Conference (WSC)*. IEEE, pp. 1226–1237.
59. Group HW et al. IEEE 1516-2010—IEEE standard for modeling and simulation (M&S) high level architecture (hla)—framework and rules. *IEEE Computer Society: Washington, DC, USA* 2010; .
60. Rasheed A, San O and Kvamsdal T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access* 2020; 8: 21980–22012.
61. Attarzadeh-Niaki SH and Sander I. An extensible modeling methodology for embedded and cyber-physical system design. *Simulation* 2016; 92(8): 771–794.
62. Jones D, Snider C, Nassehi A et al. Characterising the digital twin: A systematic literature review. *CIRP journal of manufacturing science and technology* 2020; 29: 36–52.
63. Moyne J, Qamsane Y, Balta EC et al. A requirements driven digital twin framework: Specification and opportunities. *IEEE Access* 2020; 8: 107781–107801.
64. Liu M, Fang S, Dong H et al. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems* 2021; 58: 346–361.
65. Sjarov M, Lechler T, Fuchs J et al. The digital twin concept in industry–a review and systematization. In *2020 25th IEEE International Conference on Emerging Technologies*

and Factory Automation (ETFA)*, volume 1. IEEE, pp. 1789–1796.

66. Talkhestani BA, Jazdi N, Schloegl W et al. Consistency check to synchronize the digital twin of manufacturing automation based on anchor points. *Procedia Cirp* 2018; 72: 159–164.
67. Wang L, Deng T, Zheng Z et al. Explainable modeling in digital twin. In *2021 Winter Simulation Conference (WSC)*. IEEE, pp. 1–12.
68. Zhang R, Wang F, Cai J et al. Digital twin and its applications: A survey. *The International Journal of Advanced Manufacturing Technology* 2022; 123(11-12): 4123–4136.
69. Sapkota MS. *Adaptive Simulation Modelling Using The Digital Twin Paradigm*. PhD Thesis, Bournemouth University, 2023.
70. Reifsnider K and Majumdar P. Multiphysics stimulated simulation digital twin methods for fleet management. In *2013 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. p. 1578.
71. Kober C, Adomat V, Ahanpanjeh M et al. Digital twin fidelity requirements model for manufacturing. In *2022 Proceedings of the Conference on Production Systems and Logistics: CPSL 2022*. Hannover: publish-Ing., pp. 595–611.
72. Kober C, Algan BN, Fette M et al. Relations of digital twin fidelity and benefits: A design-to-value approach. *Procedia CIRP* 2023; 119: 809–815.
73. Yue T, Ali S, Arcaini P et al. Towards requirements engineering for digital twins of cyber-physical systems. In *2022 International Symposium on Leveraging Applications of Formal Methods*. Springer, pp. 9–21.
74. Hua EY, Lazarova-Molnar S and Francis DP. Validation of digital twins: Challenges and opportunities. In *2022 Winter Simulation Conference (WSC)*. IEEE, pp. 2900–2911.
75. Sargent RG. Verification and validation of simulation models. In *2010 Proceedings of the 2010 Winter Simulation Conference*. IEEE, pp. 166–183.
76. Khan A, Dahl M, Falkman P et al. Digital twin for legacy systems: Simulation model testing and validation. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, pp. 421–426.
77. Dahmen U, Osterloh T and Roßmann J. Verification and validation of digital twins and virtual testbeds. *Int J Adv Appl Sci* 2022; 11(1): 47–64.
78. Cortés A et al. Deep air–a smart city AI synthetic data digital twin solving the scalability data problems. In *2022 Artificial Intelligence Research and Development: Proceedings of the 24th International Conference of the Catalan Association for Artificial Intelligence*, volume 356. IOS Press, p. 83.
79. ISO 23247-2:2021 Automation systems and integration — Digital twin framework for manufacturing — Part 2: Reference architecture. Standard, International Organization for Standardization, Geneva, CH, 2021.
80. Zhu Z, Liu C and Xu X. Visualisation of the digital twin data in manufacturing by using augmented reality. *Procedia Cirp* 2019; 81: 898–903.
81. Bélanger J, Venne P, Paquin JN et al. The what, where and why of real-time simulation. *Planet Rt* 2010; 1(1): 25–29.
82. Hu X. Data assimilation for simulation-based real-time prediction/analysis. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*. pp. 404–415. DOI:10.23919/ANNSIM55834.2022.9859329.
83. Huang Y, Xie X, Cho Y et al. Particle filter–based data assimilation in dynamic data-driven simulation: sensitivity analysis of three critical experimental conditions. *Simulation* 2023; 99(4): 403–415.
84. Traore MK, Gorecki S and Ducq Y. A simulation based approach to digital twin's interoperability verification & validation. In *2022 Workshop" Interoperability challenges and solutions within industrial networks" co-located with 11th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA 2022)*, volume 3214.
85. Mertens J and Denil J. Digital-twin co-evolution using continuous validation. In *2023 Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*. ICCPS '23, New York, NY, USA: Association for Computing Machinery. ISBN 9798400700361, p. 266–267. DOI:10.1145/3576841.3589628. URL https://doi.org/10.1145/3576841.3589628.
86. Beisbart C and Saam NJ. *Computer simulation validation*. Switzerland: Springer Nature Switzerland AG, 2019. ISBN 9783319707655. DOI:https://doi.org/10.1007/978-3-319-70766-2.
87. Lugaresi G, Gangemi S, Gazzoni G et al. Online validation of simulation-based digital twins exploiting time series analysis. In *2022 Winter Simulation Conference (WSC)*. pp. 2912–2923. DOI:10.1109/WSC57314.2022.10015346.
88. Overbeck L, Le Louarn A, Brützel O et al. Continuous validation and updating for high accuracy of digital twins of production systems. In *2021 Simulation in Produktion und Logistik 2021, Erlangen, 15.-17.September 2021. Hrsg.: J. Franke, Simulation in Produktion und Logistik 2021*, volume 0. Cuvillier Verlag. ISBN 978-3-73697-479-1, p. 609–617.
89. Gertler J. *Fault detection and diagnosis in engineering systems*. Boca Raton: CRC press, 1998.
90. Harada Y, Yamagata Y, Mizuno O et al. Log-based anomaly detection of CPS using a statistical method. In *2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP)*. pp. 1–6. DOI:10.1109/IWESEP.2017.12.
91. Narasimhan S and Biswas G. Model-based diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 2007; 37(3): 348–361. DOI:10.1109/TSMCA.2007.893487.
92. Hashtrudi Zad S, Kwong R and Wonham W. Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Transactions on Automatic Control* 2003; 48(7): 1199–1212. DOI:10.1109/TAC.2003.814099.
93. Ramirez-Trevino A, Ruiz-Beltran E, Rivera-Rangel I et al. Online fault diagnosis of discrete event systems. a petri net-based approach. *IEEE Transactions on Automation Science and Engineering* 2007; 4(1): 31–39. DOI:10.1109/TASE.2006.872120.
94. Li D, Wang Y, Wang J et al. Recent advances in sensor fault diagnosis: A review. *Sensors and Actuators A: Physical* 2020; 309: 111990. DOI:https://doi.org/10.1016/j.sna.2020.111990. URL https://www.sciencedirect.com/science/article/pii/S0924424719308635.
95. Abid A, Khan MT and Iqbal J. A review on fault detection and diagnosis techniques: basics and beyond. *Artificial*

Intelligence Review 2021; 54(5): 3639–3664. DOI:10.1007/s10462-020-09934-2. URL https://doi.org/10.1007/s10462-020-09934-2.

96. Darvishi H, Ciuonzo D and Rossi PS. A machine-learning architecture for sensor fault detection, isolation, and accommodation in digital twins. *IEEE Sensors Journal* 2023; 23(3): 2522–2538. DOI:10.1109/JSEN.2022.3227713.
97. Polyzotis N, Zinkevich M, Roy S et al. Data validation for machine learning. In Talwalkar A, Smith V and Zaharia M (eds.) *2019 Proceedings of Machine Learning and Systems*, volume 1. pp. 334–347. URL https://proceedings.mlsys.org/paper_files/paper/2019/file/928f1160e52192e3e0017fb63ab65391-Paper.pdf.
98. Mertens J and Denil J. The digital twin as a common knowledge base in devops to support continuous system evolution. In Habli I, Sujan M, Gerasimou S et al. (eds.) *2021 Computer Safety, Reliability, and Security. SAFECOMP 2021 Workshops*. Cham: Springer International Publishing. ISBN 978-3-030-83906-2, pp. 158–170.
99. Traoré MK and Muzy A. Capturing the dual relationship between simulation models and their context. *Simulation Modelling Practice and Theory* 2006; 14(2): 126–142.
100. Denil J, Klikovits S, Mosterman PJ et al. The experiment model and validity frame in m&s. In *2017 Proceedings of the Symposium on Theory of Modeling & Simulation*. pp. 1–12.
101. Van Mierlo S, Oakes BJ, Van Acker B et al. Exploring validity frames in practice. In *2020 International Conference on Systems Modelling and Management*. Springer, pp. 131–148.
102. Van Acker B, De Meulenaere P, Denil J et al. Valid (re-) use of models-of-the-physics in cyber-physical systems using validity frames. In *2019 Spring Simulation Conference (SpringSim)*. IEEE, pp. 1–12.
103. Otter M, Reiner M, Tobolář J et al. Towards modelica models with credibility information. *Electronics* 2022; 11(17): 2728.
104. Köhler J, Heinkel HM, Mai P et al. Modelica-association-project "system structure and parameterization"–early insights. In *2016 The First Japanese Modelica Conferences, May 23-24, Tokyo, Japan*. 124, Linköping University Electronic Press, pp. 35–42.
105. Kephart J and Chess D. The vision of autonomic computing. *Computer* 2003; 36(1): 41–50. DOI:10.1109/MC.2003.1160055.
106. Feng H, Gomes C, Gil S et al. Integration of the mape-k loop in digital twins. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, pp. 102–113.
107. David I and Syriani E. DEVS model construction as a reinforcement learning problem. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, pp. 30–41.
108. Lauer F, Bloch G, Lauer F et al. *Hybrid system identification*. Springer, 2019.
109. Singh M, Fuenmayor E, Hinchy EP et al. Digital twin: Origin to future. *Applied System Innovation* 2021; 4(2). DOI:10.3390/asi4020036. URL https://www.mdpi.com/2571-5577/4/2/36.
110. Davis PK. Exploratory analysis enabled by multiresolution, multiperspective modeling. In *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*, volume 1. IEEE, pp. 293–302.
111. Davis PK and Bigelow JH. *Experiments in multiresolution modeling (MRM)*. Rand Santa Monica, CA, USA, 1998.
112. Franceschini R, Van Mierlo S and Vangheluwe H. Towards adaptive abstraction in agent based simulation. In *2019 Winter Simulation Conference (WSC)*. IEEE, pp. 2725–2736.
113. Bosmans S, Bogaerts T, Casteels W et al. Adaptivity in multi-level traffic simulation using experimental frames. *Simulation Modelling Practice and Theory* 2022; 114: 102395.
114. Biglari R, Mertens J and Denil J. Towards Real-time Adaptive Approximation. In *2022 11th European Congress Embedded Real Time Systems - ERTS 2022*. Toulouse, France. URL https://hal.science/hal-03692186.
115. Muñoz P, Wimmer M, Troya J et al. Using trace alignments for measuring the similarity between a physical and its digital twin. In *2022 Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '22, New York, NY, USA: Association for Computing Machinery. ISBN 9781450394673, p. 503–510. DOI:10.1145/3550356.3563135. URL https://doi.org/10.1145/3550356.3563135.
116. Biglari R and Denil J. Model validity and tolerance quantification for real-time adaptive approximation. In *2022 Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '22, New York, NY, USA: Association for Computing Machinery. ISBN 9781450394673, p. 668–676. DOI:10.1145/3550356.3561604. URL https://doi.org/10.1145/3550356.3561604.
117. Mertens J and Denil J. ESS: EMF-based simulation specification, a domain-specific language for model validation experiments. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, pp. 416–427.
118. Malmodin J and Lundén D. The energy and carbon footprint of the global ICT and E&M sectors 2010–2015. *Sustainability* 2018; 10(9). DOI:10.3390/su10093027. URL https://www.mdpi.com/2071-1050/10/9/3027.
119. Anders SG and Edler T. On global electricity usage of communication technology: Trends to 2030. *Challenges* 2015; 6(1): 117–157. DOI:10.3390/challe6010117. URL https://www.mdpi.com/2078-1547/6/1/117.
120. Belkhir L and Elmeligi A. Assessing ICT global emissions footprint: Trends to 2040 recommendations. *Journal of Cleaner Production* 2018; 177: 448–463. DOI:https://doi.org/10.1016/j.jclepro.2017.12.239. URL https://www.sciencedirect.com/science/article/pii/S095965261733233X.
121. Berkhout F and Hertin J. Impacts of information and communication technologies on environmental sustainability: speculations and evidence. report to the oecd. *Science and Technology Policy Research Unit, University of Sussex, Brighton* 2001; .
122. Bellis S and Denil J. Challenges and possible approaches for sustainable digital twinning. In *2022 Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '22, New York, NY, USA: Association for Computing Machinery. ISBN 9781450394673, p. 643–648. DOI:10.1145/3550356.3561551. URL https://doi.org/10.1145/3550356.3561551.

123. Paredis R and Vangheluwe H. Towards a digital Z framework based on a family of architectures and a virtual knowledge graph. In *2022 Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. MODELS '22, New York, NY, USA: ACM, p. 491–496. DOI:10.1145/3550356.3561543.
124. Niyonkuru D and Wainer G. A DEVS-based engine for building digital quadruplets. *SIMULATION* 2021; 97(7): 485–506. DOI:10.1177/00375497211003130. URL https://doi.org/10.1177/00375497211003130. PMID: 34219819, https://doi.org/10.1177/00375497211003130.
125. Traore MK and Ducq Y. Digital twin for smart cities: An enabler for large-scale enterprise interoperability. In *I-ESA 2022 (11th international conference interoperability for enterprise systems and applications)*, volume 3214.

Author Biographies

Zeeshan Ali *is a PhD student at the University of Bordeaux, in the IMS (Integration from Material to System) laboratory.*

Raheleh Biglari *is a PhD student at the University of Antwerp, Faculty of Applied Engineering, Electronics and ICT department, Cosys-Lab.*

Joachim Denil *is an Associate Professor at the University of Antwerp, Faculty of Applied Engineering in the Electronics and ICT department, Cosys-Lab, and is associated with Flanders Make.*

Joost Mertens *is a PhD student at the University of Antwerp, Faculty of Applied Engineering in the Electronics and ICT department, Cosys-Lab.*

Milad Poursoltan *is a Post-doctoral student at the University of Bordeaux, in the IMS (Integration from Material to System) laboratory.*

Mamadou Kaba Traoré *is a Full Professor at the University of Bordeaux, with research activities in the IMS laboratory and teaching activities in the Department of Engineering Sciences.*