Digital Twins Approaches and Methods Review

Mohammed E. Helal
Faculty of Engineering, AASTMT
Cairo, Egypt
mohelal@helalsoftware.net

Hamed shawky Zied
Air Defense College, Alexandria University, Alexandria, Egypt.
Dr.hamedzied@gmail.com

Ahmed Khairy Mahmoud
Air Defense College, Alexandria University, Alexandria, Egypt.
Dr.ahmed.kairy.6470.adc@alexu.edu.eg

Manal Helal
School of Physics, Engineering & Computer Science
Hertfordshire University
HATFIELD, UK
mhelal@ieee.org

Ali E. Takieldeen
IEEE Senior Member,
Faculty of Artificial Intelligence, Delta University for Science and Technology, Gamasa 35712, Egypt
a_takieldeen@yahoo.com

Samia M. Abd-Alhalem
Faculty of Artificial Intelligence, Delta University for Science and Technology, Gamasa 35712, Egypt
Samia.mohammed@deltauniv.edu.eg

***Abstract*—This paper investigates the recent advances in Digital Twin technologies. The aim is to compare the approaches, available open source and proprietary technologies and methods, their features, and their integration capabilities. The motivation is to enable better design decisions based on the available literature and case studies. Various tools for 3D reconstruction and visualisation, IoT and sensor integration, Physical simulations and other complete platforms provide complete solutions. A conclusion of current challenges and future work identified that the lack of standardisation and interoperability makes the life-time of a digital twin short, with a high cost and time to build and rebuild if required.**

Keywords— Digital Twins, 3D reconstruction, Simulation, Causal analysis.

I. INTRODUCTION

Digital Twins (DTs) technologies emerged from flight simulations in the 2010s and can be dated back to Kalman Filters algorithms from the 60s of the last century. A DT is a digital representation of a Physical entity or Cyber-Physical System (CPS) containing a geometry model that enables 3D reconstruction to enable design optimisation, traces (timed events) for real-time monitoring, aggregated data from sensors or history for predictive maintenance and functions to control the services of the CPS and maintain and optimise it. DTs matured over the last few decades due to the availability of large deep learning models ready to work on the substantial available datasets from the widespread Internet of Things (IoT) devices and sensors, Programmable Logic Controllers (PLC), and other real-time data streams. Virtual Reality and Augmented Reality technologies are not only for games and social networks but also have widespread applications in healthcare, manufacturing, avionics, smart cities, construction, aerospace, logistics, energy and power, communications and transportation. A digital model such as in AutoCAD would simulate all requirements before actual manufacturing, and the reverse process is feasible through 3D reconstruction from images and videos. Bi-directional communication between the digital and physical twins enables continuous development and maintenance. In silico, the digital twin can simulate to produce synthetic data using known models, while, in reality, the Physical twin sends information for analysis to infer models and update knowledge about the asset being studied. Control of a Physical asset can then be optimised by choice of the most appropriate model, sensors and their placement, maintenance requirements, and feasible upgrades to avoid hazards or optimise performance [1].

This review presents the theoretical foundations required to enable DTs development, models' libraries, and available technologies, followed by use cases in various sectors. Given the availability of various tools and approaches, new adoptees of DT technologies would find it challenging to identify a suitable approach. This review aims to introduce and compare the latest advances in the field and facilitate their adoption based on the needs of a given asset type and its environment.

II. LITERATURE REVIEW

Previous surveys explained the key notations such as [2] characterised 13 key terminologies and associated processes as follows: Physical Entity/Twin; Virtual Entity/Twin; Physical Environment; Virtual Environment; State; Realisation; Metrology; Twinning; Twinning Rate; Physical-to-Virtual Connection/Twinning; Virtual-to-Physical Connection/Twinning; Physical Processes; and Virtual Processes. A display DT will only visualise the Physical asset it represents. A simulation DT requires Physical models as well to simulate the physical environment conditions around the physical asset as read from the sensors attached. While a Causal DT further models the interactions between various DTs and their environment from acquired sensor data about features and states to identify cause-effects mapping [3]. The different DT Types can be thought of as hierarchical in nature in terms of what technologies each type requires to build it. The Display DT is the base type that needs to be built using 3D geometry methods and visualisation platforms. The simulation DT builds on top of it defining functions and processes. Finally, the Causal DT will use various methods from ML and control algorithms to predict and optimise the DT. This is illustrated in Figure 1.

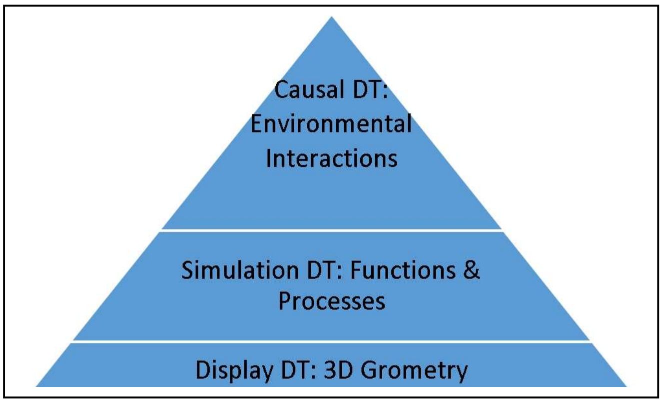

Figure 1: DT Types and their requirements

A product life-cycle for building a DT project is identified in [4] containing stages of Imagine, Define, Realise, Support/Use, and Retire/Dispose. The Imagine and Define phases received less attention in recent research endeavours. Although, if not appropriately planned, considerable time and cost can be wasted in implementing detailed functionality that could have been best approached using different platforms, technologies and algorithms. CPS domain experts would specify a choice of functional requirements, along with various properties of the environment and possible data communication details in the Define stage. In the Realise stage, a main concern is data ownership and security, whether in the communication protocol choice and setup or the data storage and shared access. Another concern in the Realise stage and its testing phase is the performance metrics required to evaluate the performance of the DT. A simulation DT requires the highest levels of fidelity with the CPS, such as the simulation environment, and its results precisely measure the physical environment. Performance metrics include network, computing speeds, and ML model accuracy in estimating the CPS state vector or predicting future states. For example, a high-fidelity DT model can be useful in forensics when aggregated data are shared with insurance companies to investigate a car accident's causes, and DTs prediction matches physical environmental conditions. More performance measures are discussed in the conclusion section. Identifying the end of life of a DT model to retire and dispose of it happens when a malfunction is identified that can not be fixed, or a required improvement or feature can not be implemented using DTs current technologies. The ongoing interoperability research endeavour might enable a plug-and-play of different tools and technologies when required. A final decision on migration to another platform or technology might be possible with new tools as they come up. This might create a cycle from the define stage, then realise, support, and then retire/migrate stage, as illustrated in Figure 2.

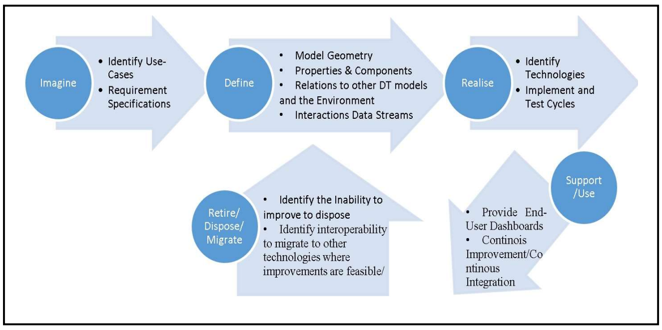

Figure 2: A DT LIfe Cycle Design Decisions with Possible Continous Integration/Continous Development

From various references [5]–[8], it does not seem that DT development has its own standards from any standardisation bodies as of the time of writing this review. However, a joint subcommittee from the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC), ISO/IEC JTC 1/SC 41 is currently developing a standard specific to DT technologies.

DT development evolved from various existing technologies, each with its own standards, such as IoT reference architecture ISO/IEC 30141, and domain-level standards, such as ISO TC 184/SC4 for Automation/Integration of industrial data for SO TC 184/SC4. Many other components have their own standards, such as systems modelling, security, communication protocols, data formats, and visualisation, among others.

Theoretical Foundations:

The same theoretical foundations for machine learning (ML) are required to enable DT Technologies at the third level of Causal DTs and predictive control. As identified in [9], the required foundation range from basic probability theory, linear algebra, optimisation, various ML algorithms and deep learning architectures and models, differential equations, stochastic differential equations, Inverse Problems, finite element analysis and uncertainty quantification using Bayesian models or Kalman Filters. The book reviews these concepts in enough detail required to get started with example code from various programming languages and environments and also references other books from which more depth or details can be studied.

All traditional AI models, and recent advances in Artificial Neural Networks (ANN) with their various complex architectures, learn from data (observations or features) a mapping function to a required target. If the features are effects and the target is the cause of these effects, this is known as Causality or causal analysis. Causal analysis is mathematically formulated in various approaches. IoT devices and sensors provide many environmental observations related to the physical asset being simulated by a DT. Studying these observations to identify the cause of a particular effect (such as the state vector of the asset) is an inverse problem formulated in the opposite direction (effects to causes), which is the reverse of the direct mapping (causes to effects). Bayesian modelling and Kalman Filters are widely adopted in DTs. A Graphical model approach has been applied in [3] to enable prediction and control at a scale between digital and Physical assets. Kalman Filter Algorithm has been used for causal analysis of DTs in [10]. In comparison, a Recurrent Neural Network model has been employed in [11], producing similar estimations to those provided by Kalman Filter models, particularly for linear models.

Also, Adaptive Control Systems such as Fuzzy Logic, ANN, Evolutionary Algorithms, and Complex Systems are useful for Data-Driven Control as opposed to Model-Based Control discussed next [12]. The technologies compared below would use the mathematical model of each algorithm when needed without re-implementing from scratch.

Digital Twin Data (DTD) is a term coined with DT-related data to manage various processes such as comprehensive data gathering, storage, mining, multi-modal fusion and association, interaction, iterative optimisation, servitisation (make available for a fee), universality, and on-demand usage. This data is classified into six types: physical entity-related data, virtual model-related data, service-related data, fusion data, connection data, and domain knowledge [13].

Models Libraries

A DT can be derived from a user-derived model, matching the nearest model from a model library or by being data-driven entirely. The various IoT cloud providers allow users to define their DTs using various interfaces and modelling languages. MS Azure, for example, uses a Digital Twin Definition Language (DTDL) to enable users to define their own models in their own vocabulary, using existing models to inherit from or interact with drawing a graph of DTs. The work in [1] compared the modelling languages of MS Azure, AWS and Eclipse DT platforms and attempted standardising them with Object Oriented Unified Modelling Language (UML) to facilitate interoperability and migration. Other platforms' modelling processes will be reviewed in the next section.

Physical Models of the services performed by the CPS must also be integrated with the DT model. This requires close collaboration between the DT developer and the CPS expert.

For example, a car DT will require kinematic engines to simulate a driving mechanical engine model to simulate the engine dynamics and maintenance requirements, among many other sensors and IoT devices that can be attached to provide real-time control and optimisation. A review of Model-based predictive control mathematical and physical literature is found in [14]. A DT comprehensive platform is ideally equipped with MultiPhysics simulation environments with a common solver for mechanical, electrical, structural, chemical, and electronic processes. Alternatively, subsystems are addressed to reduce the model complexity. Simulation software includes COMSOL, SimScale, AnyLogic, Ansys and MATLAB® and Simulink [15].

Available Technologies

3D reconstruction and visualisation are fundamental technology for DT building. 3D data will likely blend architecture, engineering, product, process, and Geospatial data, as well as important metadata. Existing models can be inherited from various computer-aided design (CAD), computer-aided engineering (CAE), Product Life Cycle Management (PLM), maintenance, repair and operations (MRO), Geographic information systems (GIS), Building Information Modeling (BIM), robotics models among others. Universal Scene Description (USD) is an open-source 3D scene description that captures 3D scene contents for creation and interoperability between the different tools. Hiring Digital Artists to create a digital model from scratch sometimes is optimal for some developments.

Data streams and analytics tools are well-standardised now. There are IoT Cloud providers, open-source to build own servers, and proprietary solutions for IoT Devices and systems. Simulation software has been used for decades for in silico simulation of expensive physical systems. Some are proprietary, and some are open source. Other tools and technologies are available for use and well documented, such as data streaming technologies like Kafka, Apache Storm and various databases for different objectives, visualisation and exploratory data analysis, machine learning models and pre-trained models. Many studies identified pipelines of existing tools to provide the required functionality, and others produced specific methods for one of the functions.

Use Cases

DT has been built for various individual CPSs, such as people, cars, buildings, biological cells, and machines. Environmental monitoring requires a number of DTs interacting together, such as a fleet of cars, roads and their contents such as buildings, cars and so forth, smart energy grids showing demand and supply entities, and smart agriculture. Processes as well are represented as functions of hypothetical DTs, such as manufacturing processes, micro- and macro-economics functionalities by the different entities, citizen science, and Industrial Symbiosis (IS). IS monitor various industries' circular use of resources such as water, energy, and the production of materials and byproducts to guarantee sustainable development [16].

III. METHODOLOGY

This review aims to identify the steps to get started with DT technologies and compare existing platforms, tools, libraries and algorithms that can offer the required functionalities. Some identified platforms and tools are open-source such as OpenTwins, PhotoScene, and Eclipse. Other proprietary platforms exist, such as Matlab, Nvidia, Microsoft, Amazon Oracle, IBM Watson IoT, Siemens MindSphere, PTC ThingWorx, ScaleOut, Dassault Systèmes, Dassault Systèmes 3DEXPERIENCE, GE Predix, and SAP Leonardo. The review includes the modelling approach, features, integration, and interoperability with other required technologies. A search was conducted through recent publications, books, and online publications to identify the conclusions of current studies, best practices, and features. For brevity, not all existing technologies are reviewed. A selection of ready-to-use well-founded platforms is reviewed below. The first three are open source, while the rest are proprietary.

1. Eclipse Foundation DT suite

The Eclipse Foundation provides an open-source Digital Twin (DT) suite called Eclipse Ditto. It offers a framework and tools for building and managing digital twins.

DT Modelling: Eclipse DT Suite is an open-source platform that consists of Vorto text tool for modelling DTs using VortoLang modelling language (a metamodel based on the Eclipse Modeling Framework (EMF)). VortoLang is composed of InformationModels that define the DT properties and FunctionBlocks that define the Operation, Event, configuration, fault, and status information.

Other Services: Eclipse Ditto enables RESTful web services to the DT, and Hono or Bosch IoTHub enables device connections using various protocols. These tools can be installed on privately owned and/or self-hosted web servers.

Use cases built using the Eclipse DT suite is diverse. Some organisations that have expressed interest in or contributed to the Eclipse Ditto project include Bosch, Siemens, SAP, Deutsche Telekom, and Vodafone. These organisations have a strong presence in manufacturing, energy, telecommunications, and software development industries.

2. OpenTwins

This is an open framework based on Eclipse Ditto DT modelling platform integrated with other open source frameworks for Unity 3D visualisation, MongoDB for DT Data Storage, InfluxDB for Time-series Data, Kafka data streaming, Grafana data visualisation, Kafka ML framework and other planned to integrate in the future.

3. PhotoScene

PhotoScene converts an image into a renderable 3D scene using a collection of algorithms in a pipeline [17].

DT Modelling: A DT is modelled from images that are then segmented to identify the objects it contains and textures to estimate scene normals. The scene normals produce a coarsely aligned 3D scene model. Using MaskFormer, the material of the objects is calculated. Generally, texture smoothing is performed, but PhotoScene uses 71 graphs from MATch tool to find the nearest match to the materials in the input image using VGG CNN weights to improve the material.

4. Matlab

Matlab is the Mathworks matrix algorithms platform rich in toolboxes for various robot manipulation, Physics models, simulations, Data visualisation, and ML models. Matlab does not offer a specific DT suite. However, Matlab is a widely used software tool in various domains, and it is utilised to build digital twin models and perform simulations and analysis

DT Modelling: Matlab has toolboxes for image processing and computer vision that enable 3D reconstruction that can be used for building DT models and their environments.

Other Services: Matlab offers diverse algorithms and toolboxes, such as modelling, simulation, control, optimisation, data analytics, data-driven ML models and ANNs. Also, physics-based modelling is enabled by Simulink to attach mechanical, hydraulic, and electrical components.

Matlab can integrate with IoT cloud providers over various streaming protocols such as Kafka.

Matlab has been employed in numerous DT development applications in aerospace and defence, automotive, energy, manufacturing, and healthcare domains. Organisations using Matlab include BMW and Ford for automotive DTs and General Electric (GE) for energy-related DT models. Also, Siemens used Matlab to build DTs for assembly lines and production workflows to analyse and improve system performance, reduce downtime, and enhance overall efficiency. ABB used Matlab to create virtual models of robotics for real-time automation and control. Bosch used Matlab to build DTs for CNC machines (Computer Numerical Control) used for precision machining operations and assembly systems.

Siemens Healthineers used Matlab to build DTs for medical imaging devices such as X-ray, computed tomography (CT), and magnetic resonance imaging (MRI) systems. They integrate physical system parameters, sensor data, and imaging algorithms to simulate and optimise imaging workflows.

5. Nvidia Omniverse

Nvidia is another proprietary hardware and software provider that produced Omniverse with owned platforms.

DT Modelling: Omniverse is founded on USD 3D standards that can migrate 3D assets from various sources that can scale well. Robot models can also be used from NVIDIA Issac Sim. The foundation of NVIDIA Omniverse is built upon NVIDIA's high-performance graphics processing units (GPUs) from the RTX series. Material Definition Language (MDL) is NVIDIA's language for describing and defining materials in a physically based manner. MDL facilitates the accurate representation of light interactions, reflections, and other optical characteristics. NVIDIA OptiX is a ray tracing framework used within Omniverse to simulate light and generate photorealistic renderings.

Other Services: Omniverse provides real-time visualisation, simulation, and collaboration capabilities. It also offers a development environment to build apps and microservices. Physics models for simulation can be built using Physx open source, migrated from open source NVIDIA Modulus, or other physics solvers such as SimScale and Kitware ParaView. Other valuable platforms include NVIDIA Metropolis for computer vision and ML APIs.

Use cases of DT models built using NVIDIA Omniverse vary across various domains such as Architecture, Engineering, and Construction (AEC), Autonomous Vehicles and Robotics, Manufacturing and Industrial Processes, Smart Cities and Urban Planning, and Energy and Utilities. NVIDIA has partnerships with several automotive companies, such as BMW, Volkswagen Group, Audi, Toyota, and Volvo, but not clear if they used it to build automotive DTs.

6. AWS IoT TwinMaker

Amazon Web Services (AWS) enable developers to build DT models of physical devices or assets, making monitoring, managing, and optimising those assets remotely more accessible.

DT Modelling: AWS IoT TwinMaker enables DT model specification using APIs and JSON REST requests. The model is defined as a ComponentType with properties as tags and PropertyDefinitions. Functions are defined as well using DataConnector to function implementation.

Other Services: Data communication is enabled over RESTful web services and APIs. The platform also enables 3D visualisation of the CPS and its physical space. AWS Greengrass enables connections to various devices, and custom dashboards can be designed for end-users. AWS platform includes key services such as Real-time data ingestion and analysis, Predictive analytics and machine learning algorithms, and integration with other AWS services, including IoT Core, Lambda, and S3.

Use cases of DTs built using AWS technologies include applications in predictive maintenance for industrial machinery and equipment, remote monitoring and control of smart buildings and infrastructure, optimisation of energy usage and resource allocation, and real-time tracking and monitoring of vehicles, assets, and supply chains. Example client organisations using AWS IoT TwinMaker include Siemens, which used AWS IoT TwinMaker to create a digital twin of a gas turbine to optimise performance and reduce maintenance costs. Shell used AWS IoT TwinMaker to develop a predictive maintenance solution for offshore oil and gas platforms. Philips Lighting used AWS IoT TwinMaker to create digital twins of streetlights for remote management and optimisation.

7. MS IoT Hub and Azure DTs

Microsoft Azure IoT Hub and Azure Digital Twins (ADTs) utilise a combination of technologies, tools, and algorithms to build and manage Digital Twins.

DT Modelling: ADT cloud services offer DTDL language that is based on JSON format and a comprehensive set of APIs and tools. DTDL allow defining the model from scratch or inheriting from another. A model is defined by name, ID, and other properties. A DT model can have relationships to other models to exchange data, attached components as other models, and commands request and response.

Other Services: Connection to devices can be performed using the Azure IoT Hub. Azure IoT Hub is a cloud-based service that acts as a central message hub for bi-directional communication between devices and the cloud. It provides secure and scalable device connectivity, data ingestion, and command/control functionalities for IoT solutions. IoT Hub supports protocols such as MQTT, AMQP, and HTTPS for device communication. Azure IoT Edge extends the capabilities of Azure IoT Hub by allowing the processing and analysis of data at the edge devices themselves. This is beneficial for scenarios where low latency, offline capabilities, or reduced cloud dependency are required. Azure Functions is a serverless computing service that allows running code snippets or functions in a scalable manner. It can be used to implement custom business logic and event-driven processing for IoT solutions. The platform also enables 3D visualisation of the CPS and its physical space. Data analysis is provided using various services such as Azure Time Series Insights, Stream Analytics, Machine Learning, Cognitive Services and Advanced analytics and algorithms [18].

Use cases of DTs built using MS Azure DT technologies span various domains such as Predictive Maintenance, Smart Buildings, Smart Cities, Energy, Agriculture, Manufacturing, Asset Tracking and Management. MS IoT Hub and Azure Digital Twins (DTs) clients include Johnson Controls in their OpenBlue platform for smart buildings; Thyssenkrupp created DTs for their elevator systems; Bentley Systems iTwin platform created DTs for infrastructure assets; ICONICS built DTs for industrial equipment and processes. Ecolab built DTs for water management solutions. Automotive clients for MS DT technologies might include Renault-Nissan-Mitsubishi Alliance, Volkswagen Group, BMW Group, and Volvo Cars.

IoT Cloud Providers: Various IoT Cloud Providers comparisons are published in papers and blogs such as [19] and [20] can help identify the suitable provider for a given use-case based on features, cost, interoperability and standardisation. Alternatively, building self-managed servers is always available and suitable for secured data storage and interoperability with existing systems.

In summary, there are four major platform providers in DT technologies. Eclipse Foundation Digital Twin (DT) suite is an open-source solution, MATLAB is a versatile numerical computing environment, NVIDIA Omniverse focuses on 3D simulation and visualisation, and Microsoft Azure IoT Hub and Azure Digital Twins provide a comprehensive cloud-based platform for IoT device management and digital twin development. The choice among these tools depends on the users' specific requirements, preferences, and expertise in the respective domains. Table 1 attempts to provide a non-exhaustive summary of the major industrial organisations in the various domains and their choices of DT providers. The table identifies Matlab as favoured by manufacturing and industrial organisations, Nvidia Omniverse as favoured by Automotive organisations, and MS Azure for smart buildings. As an open-source provider, Eclipse is favoured by telecommunication organisations, while some manufacturing organisations are interested in it. The information available in this table is collected from the public domain. Further details can be collected from the DT platform providers and/or the CPS organisations.

Table 1: DTs Providers and Major Application Industries

		ECLIPSE FOUNDATION	MATLAB	NVIDIA OMNIVERSE ENTERPRISE	aws	Microsoft Azure
Manufacturing	BOSCH	✓	✓			
	SIEMENS	✓	✓		✓	
	GE		✓			
	ABB		✓			
	PHILIPS				✓	
Teleco	vodafone	✓				
	T	✓				
Automotive	BMW		✓	✓		✓
	Ford		✓			
	VW			✓		✓
	Audi			✓		
	TOYOTA			✓		
	VOLVO			✓		✓
						✓
Softwar	SAP	✓				
Energy	Shell				✓	
Smart Building	Johnson Controls					✓
	Bentley					✓
	thyssenkrupp					✓

One example of a DT of a car is the Z1 Dashboard software that is used by many manufacturers, including Porsche, in their 911 GT3 Cup Car simulator. The digital twin includes a full vehicle model with all components, including the engine, transmission, suspension, and aerodynamics, and allows users to simulate and optimise the performance of the car in various scenarios. Generally, Porche uses DTs collected data to improve new generations of their cars [21], [22]. Another example of a DT for a smart city is Amsterdam, which is made available through a cell phone application [23].

IV. CONCLUSION

Depending on their specific application and objectives, DTs can be evaluated against various performance metrics that the CPS domain expert needs to further specify in their requirements. Ten proposed metrics can be used in the evaluation. The *effectiveness and efficiency* of a DT can be assessed by the accuracy of its ability to replicate the behaviour and characteristics of the physical system. This includes comparing the data generated by the digital twin with real-time data from the physical system to ensure alignment. The *reliability* of a DT can be assessed by its availability when needed to provide consistent and uninterrupted insights and analysis, such as measuring its uptime. A DT *responsiveness* evaluates its ability to provide real-time or near real-time insights is crucial. Evaluating its responsiveness involves assessing the latency or delay in data transfer, analysis, and feedback between the physical system and the digital twin. A DTs *scalability* metric evaluates its ability to scale to accommodate larger systems or networks and handle increased data volumes. Evaluating scalability helps ensure that a DT can handle growth and expansion without compromising performance. A DT's *computational efficiency* measures the computational resources required to operate the digital twin. Evaluating computational efficiency helps identify potential bottlenecks and optimise resource allocation, ensuring the digital twin operates cost-effectively. The *predictive capability* of a DT measures the effectiveness of its forecast system behaviour and makes proactive recommendations. Assessing the accuracy and reliability of the digital twin's predictions against real-world outcomes helps measure its predictive capability.

The maintenance and update efficiency of a DT measures the required periodic updates and maintenance to remain effective. Evaluating the ease and efficiency of updating and maintaining the digital twin helps it adapt to changes in the physical system or underlying technology. *The cost-effectiveness* of a DT considers the overall cost of developing, implementing, and operating the digital twin compared to its value. Evaluating cost-effectiveness helps determine the return on investment (ROI) and the economic viability of the digital twin solution. *Security* of a DT deal with sensitive data and must maintain a high level of security. Evaluating the digital twin's security measures and vulnerability to cyber threats helps protect critical systems and data. Finally, the *user experience* of interacting with a DT should be evaluated. This includes assessing the intuitiveness of the interface, ease of data interpretation, and the availability of relevant visualisations and actionable insights [24].

As reviewed in the previous sections, planning a DT project is challenged by many design decisions in all stages, and each affects the success, serviceability, and longevity of the end product. The various platforms, tools, and algorithms make the design decisions available for a new project vary from one use case to another. A choice might be more costly or not enabling future enhancements, integration with other tools, or required new features. The same software engineering methods for any computing development project are required to enable best practices of low coupling between components and high coherence within components' contents, reusability of code, and change management. Interoperability and standardisation are still an open research gap in the current state of the art of DT technologies. An initial choice for a project might not be changed later without a considerable loss of resources.

Future work will focus on testing these technologies using a few case studies, such as building a Digital Twin for a building or a car and comparing their performance in different development platforms. Research gaps exist on the wide adoption of these technologies to provide different synthetic data/real data integration approaches for various application domains and control scenarios. A detailed conclusion needs to be drawn on how intuitive the process is using the various technologies and how easy to integrate the created models with other tools such as bidirectional communication, 3D reconstruction, visualisation, and ML algorithms for analysis and predictions. Also, working towards current research gaps is planned, such as testing the various platforms on integrating various virtual entities to provide more realistic environments interactions rather than the simple models that ignore many physical factors in the environment. Once the most suitable platform is identified, a DT model can be built to apply innovative use cases for various objectives, such as reducing the cost of design, maintenance, reconfiguration, and risks of simulating new functions and interactions

References

[1] J. Pfeiffe, D. Lehner, A. Wortmann, and M. Wimmer, 'Modeling Capabilities of Digital Twin Platforms - Old Wine in New Bottles?', *J. Object Technol.*, vol. 21, no. 3, p. 3:1, 2022, doi: 10.5381/jot.2022.21.3.a10.

[2] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, 'Characterising the Digital Twin: A systematic literature review', *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020, doi: 10.1016/j.cirpj.2020.02.002.

[3] M. G. Kapteyn, J. V. R. Pretorius, and K. E. Willcox, 'A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale'. arXiv, Apr. 13, 2021. Accessed: Mar. 27, 2023. [Online]. Available: http://arxiv.org/abs/2012.05841

[4] R. Stark and T. Damerau, 'Digital Twin', in *CIRP Encyclopedia of Production Engineering*, S. Chatti and T. Tolio, Eds., Berlin, Heidelberg: Springer, 2019, pp. 1–8. doi: 10.1007/978-3-642-35950-7_16870-1.

[5] F. Flamigni, P. Pileggi, O. Barrowclough, and J. Haenisch, 'FIRST REPORT ON STANDARDS RELEVANT FOR DIGITAL TWINS', The Change2Twin Consortium, 2020.

[6] K. Wang, Y. Wang, Y. Li, X. Fan, S. Xiao, and L. Hu, 'A review of the technology standards for enabling digital twin', *Digit. Twin*, vol. 2, p. 4, Nov. 2022, doi: 10.12688/digitaltwin.17549.2.

[7] A. Kung, C. Baudoin, and K. Tobich, 'Report of TWG Digital Twins: Landscape of Digital Twins', Zenodo, May 2022. doi: 10.5281/ZENODO.6556917.

[8] K. E. Harper, S. Malakuti, and C. Ganz, *J. Innov. IIC*, 2019, doi: 10.25607/OBP-1860.

[9] M. Asch, *A Toolbox for Digital Twins: From Model-Based to Data-Driven*. in Mathematics in Industry. Society for Industrial and Applied Mathematics, 2022. doi: 10.1137/1.9781611976977.

[10] D. P. Madhavan, 'Stochastic Formulation of Causal Digital Twin – Kalman Filter Algorithm', May 2021, [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2105/2105.05236.pdf

[11] D. P. Madhavan, 'Evidence-based Prescriptive Analytics, CAUSAL Digital Twin and a Learning Estimation Algorithm', Apr. 2021, doi: https://doi.org/10.48550/arXiv.2104.05828.

[12] R. He, G. Chen, C. Dong, S. Sun, and X. Shen, 'Data-driven digital twin technology for optimised control in process systems', *ISA Trans.*, vol. 95, pp. 221–234, Dec. 2019, doi: 10.1016/j.isatra.2019.05.011.

[13] M. Zhang *et al.*, 'Digital twin data: methods and key technologies', *Digit. Twin*, vol. 1, p. 2, Sep. 2021, doi: 10.12688/digitaltwin.17467.1.

[14] D. Q. Mayn, 'Model predictive control: Recent developments and future promise | Elsevier Enhanced Reader', *Elsevier Autom.*, vol. 50, pp. 2967–2986, 2014, doi: 10.1016/j.automatica.2014.10.128.

[15] N. Khaled, B. Pattel, and A. Siddiqui, *Digital twin development and deployment on the cloud: developing cloud-friendly dynamic models using Simulink/Simscape and Amazon AWS*. London, United Kingdom ; San Diego, CA: Academic Press, an imprint of Elsevier, 2020.

[16] A. Kamilaris, V. Wohlgemuth, K. Karatzas, and I. Athanasiadis, Eds., *Advances and New Trends in Environmental Informatics: Digital Twins for Sustainability*. in Progress in IS. Cham: Springer, 2021.

[17] Y.-Y. Yeh *et al.*, 'PhotoScene: Photorealistic Material and Lighting Transfer for Indoor Scenes', in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 18541–18550. doi: 10.1109/CVPR52688.2022.01801.

[18] A. Meijers, *Hands-On Azure Digital Twins*. Packt Publishing, 2022.

[19] T. Pflanzner and A. Kertesz, 'A survey of IoT cloud providers', in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia: IEEE, May 2016, pp. 730–735. doi: 10.1109/MIPRO.2016.7522237.

[20] '10+ Best IoT Cloud Platforms', *Euristiq*. https://euristiq.com/best-iot-cloud-platforms/ (accessed Mar. 30, 2023).

[21] 'Z1 Sim Dashboard | Porsche GT3 Cup Car Simulator'.

https://www.z1simwheel.com/dashboard/porsche.cfm (accessed May 28, 2023).
[22] 'A digital chassis twin for predictive driving functions and component status updates', *Porsche Newsroom*, Nov. 04, 2021. https://newsroom.porsche.com/en/2021/innovation/porsche-digital-chassis-twin-26061.html (accessed May 28, 2023).
[23] Amsterdam Smart City, 'Amsterdam Innovation Tour (self-guided via app)', *Amsterdam Smart City*, Jul. 04, 2018. https://amsterdamsmartcity.com/updates/experience/5-amsterdam-innovation-tour-self-guided-via-app (accessed May 28, 2023).
[24] S. S. Kamble, A. Gunasekaran, A. Ghadge, and R. Raut, 'A performance measurement system for industry 4.0 enabled smart manufacturing system in SMMEs- A review and empirical investigation', *Int. J. Prod. Econ.*, vol. 229, p. 107853, Nov. 2020, doi: 10.1016/j.ijpe.2020.107853.