Understanding Digital Twins for Cyber-Physical Systems: A Conceptual Model*

Tao Yue[1,2][0000−0003−3262−5577], Paolo Arcaini[3][0000−0002−6253−4062], and Shaukat Ali[2][0000−0002−9979−3519]

[1] Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2] Simula Research Laboratory, Oslo, Norway
[3] National Institute of Informatics, Tokyo, Japan

Abstract. Digital Twins (DTs) are revolutionizing Cyber-Physical Systems (CPSs) in many ways, including their development and operation. The significant interest of industry and academia in DTs has led to various definitions of DTs and related concepts, as seen in many recently published papers. Thus, there is a need for precisely defining different DT concepts and their relationships. To this end, we present a conceptual model that captures various DT concepts and their relationships, some of which are from the published literature, to provide a unified understanding of these concepts in the context of CPSs. The conceptual model is implemented as a set of Unified Modeling Language (UML) class diagrams and the concepts in the conceptual model are explained with a running example of an automated warehouse case study from published literature and based on the authors' experience of working with the real CPS case study in previous projects.

Keywords: Cyber-Physical Systems · Digital Twins · Conceptual Model

1 Introduction

Cyber-Physical Systems (CPSs) are complex interdisciplinary systems present in many domains. With time, these systems are getting even more complicated due to, e.g., ever-changing hardware, software updates, protocols, the increased use of advanced artificial intelligence (AI) techniques, and highly uncertain operating environments. As a result, advanced technologies such as digital twins (DTs) have been proposed to ensure correct development and dependable operation throughout lifetimes of such CPSs.

However, DTs are not specific to CPSs and can be built for any living or non-living thing. For example, there exists an initiative in the European Union

* The work is supported by the National Natural Science Foundation of China under Grant No. 61872182. The work is also partially supported by the Co-evolver project (No. 286898/F20) funded by the Research Council of Norway under the FRIPRO program. Paolo Arcaini is supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST; Funding Reference number: 10.13039/501100009024 ERATO.

about building digital twins for life sciences[4] and even building a digital twin of the Earth (i.e., the Destination Earth (DestinE) initiative)[5]. In the context of this paper, we focus only on DTs for CPSs. DTs for CPSs may share similarities with DTs for other things, but may also differ. For instance, a DT built for an ocean may be reused for various CPSs operating in the ocean such as subsea oil production systems [6] and autonomous vessels [11].

DTs for CPSs have attracted huge interests in industrial and academic circles, as it can be seen by major companies promoting their digital twin platforms (e.g., ANSYS[6], Siemens[7]), and an increased number of scholarly publications [16,21,25]. Due to the increasing interest in DTs for CPSs, DTs and their concepts have been emerging with various definitions. Though some recent papers have started to unify the meaning of such concepts (e.g., [16,22,25]), due to several similar publications appearing close to each other, they have resulted in another set of definitions of DTs and their concepts. To this end, we present a conceptual model for the purpose of unifying various DT concepts. In addition, in our conceptual model, we made the effort to establish relationships among the concepts and characterize important concepts such as DTs with their functionalities, maturity level, and quality requirements. We developed our conceptual model as a set of UML class diagrams. To explain the concepts, we used a running example of a CPS case study from the logistics domain.

The rest of the paper is organized as follows. Section 2 presents the description of our running example. Section 3 introduces core concepts, while Section 4 describes more detailed concepts of a DT and its CPS. We discuss evolution and uncertainty of DTs and their CPSs in Section 5 and their life-cycles in Section 6. Finally, we present related work in Section 7 and draw conclusions in Section 8.

2 Running example

To explain various concepts, we will use the running example of a CPS case study, i.e., Automated Warehouse System (AWS), all throughout the paper to explain the conceptual model. This AWS CPS case study has been used in our previous works for assessing the definition of: a conceptual model for uncertainty in CPSs [30], a UML state machine based methodology for modeling uncertainty [29], a model-based testing framework for CPS testing under uncertainty [28], a CPS product line modeling methodology [23], and an approach for monitoring CPSs [15].

AWSs can be used for diverse applications, such as in food industry for automatically producing orders for supermarkets. An AWS system typically consists

[4] https://euroocs.eu/funding-opportunities/eu-fet-proact-eic-07-2020-digital-twins-for-the-life-sciences/

[5] https://ec.europa.eu/digital-single-market/en/destination-earth-destine/

[6] https://www.ansys.com/products/systems/digital-twin

[7] https://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465

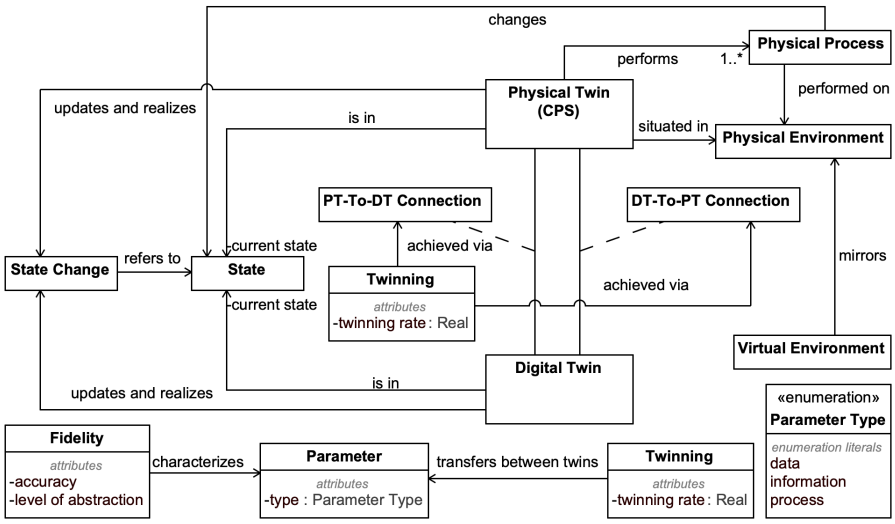

Fig. 1. Physical Twin, Digital Twin, their Connections and Environments

of many different physical systems such as conveyor belts, sorting systems, and cranes. An AWS usually has an accompanying supervision system, which is often deployed on a cloud. This supervision system performs various operations such as monitoring warehouses, taking actions in case of emergency situations, and performing advanced analyses based on data collected from various warehouses. This can be considered as the initial version of a DT for the AWS. We can imagine that the DT can be extended with additional functionalities to reach a higher level of maturity (see Section 4.2). In this paper, we describe our vision towards this direction, and illustrate it with this running example.

3 Overview

In this section, we introduce basic concepts related to DTs. We introduce common concepts that have been used in different papers, as reported in the systematic literature review in [16]. Note that sometimes different terms have been used in different papers to refer to the same concept. We will report the most common term or, in any case, the one that we think is more representative. Figure 1 reports the concepts we identified.

A central concept is *Physical Twin* (also called *physical entity*), representing the physical artifact for which a DT has been built or is to be built. Examples of physical twins include buildings, vessels, farms, and vehicles. In our context, we particularly focus on CPSs, and we use the example of an AWS in this paper. Various concepts related to the DT of AWS are listed in Table 1.

A physical twin (PT) is characterized by a list of *physical process*es that it performs (see Figure 1). Different processes require different considerations by a

DT. Some are critical processes required to be monitored by the DT, while others are non-critical and no attention is needed from the DT. Usually, the execution of a monitored process produces a state change, which should be observed and dealt with accordingly by the DT. A PT exists and lives/operates in a *PT environment* that encompasses all the elements that can affect, in some way, its operation. Examples are environmental conditions of the AWS such as temperature and humidity in the warehouse. More detailed discussions about PTs are provided in Section 4.1.

Digital Twin (also named *virtual twin*) is the central concept of the conceptual model we propose in this paper. A DT of a PT represents the digital and live replica of the PT. The DT synchronizes with its PT, and performs different functionalities. More detailed discussions about *DT Functionalities* are provided in Section 4.2.

A DT interacts with a *virtual environment* (see Figure 1), which is a replica of the physical environment of its counterpart PT. Note that some of the aspects of the physical environment may be unknown, and so the view provided by the virtual environment is, by definition, partial. Please refer to Section 5 for more detailed discussions on this. Having the virtual environment separated from the DT is convenient, as this could be reused over different DTs (e.g., the virtual environment of the warehouse could be used for the DT of a food production and retail). Moreover, variations of the virtual environment could also be used for prediction activities, to predict what could happen to the PT under possible future (probably uncertain) environmental conditions.

Both the PT and the DT are characterized, at any given time point, by their *State* (see Figure 1). The twins operate by changing their states. The concept of *state change* is particular relevant in this context, as the modification of the current state of the PT could be used to trigger analyses implemented in the DT.

An important concept for the operation of the DT is the connection between the twins. This is actually achieved through two different types of connections: *PT-To-DT-connection* and *DT-To-PT-connection* (see Figure 1). PT-To-DT-connection allows the DT to update its view of the state of the PT (e.g., the positioning of stock and number of shelving units), i.e., transmitting such information through the connection from the PT to the DT. In the opposite direction, the DT-To-PT-connection allows the DT to trigger the evolution of the operating PT for, e.g., applying some mitigation actions. For instance, the DT of an AWS might send a request to reboot one or more programmable logic controller (PLCs) of an AWS system. Another example could be that the DT of a smart building [10] could request its Heating, Ventilation, and Air Conditioning (HVAC) system to turn down the temperature of some empty rooms, as a shortage of electricity has been forecasted by a prediction model of the DT. The connection can be established with the same connection means used in the PT-To-DT-Connection. Additionally, the DT-To-PT-connection requires that the PT provides actuators to apply changes decided by the DT. Note that the DT-To-PT-Connection could also be performed by a human operator who,

on the basis of the outcomes of the analyses performed by the DT, manually modifies the PT. This is necessary when no digital connection can be established, and/or the PT does not provide a way to be digitally controlled. For instance, an operator of the DT of an AWS system might directly talk to an onsite AWS operator to inspect a malfunctioning PLC in the warehouse. Alternatively, when the DT observes an anomaly of a PLC, the DT sends a signal directly to the computer that a specific AWS operator is interacting with and requests for manual inspection of the PLC.

The overall process of synchronization (in both directions) between the PT and the DT is referred to as *Twinning* (see Figure 1). The twinning is characterized by the *twinning rate*, i.e., how often the synchronization takes place. In case of (almost) real-time synchronization, the twins are always aligned. However, if the twinning rate is non-negligible, there could be moments in which, for example, the DT has an outdated view of the PT.

The type of information that is exchanged between the twins can be described in terms of *parameter*s, each characterized by a *parameter type* (see Figure 1). The number and the accuracy of these parameters define the *fidelity* of the virtual representation. Indeed, despite the name *twin*, the DT is by definition an *abstraction* of the PT. Increasing the number and accuracy of the parameters allows to lower down the abstraction and, therefore, increase the fidelity of the representation.

The running case study illustrating this part of the conceptual model is shown in Table 1.

4 Physical twin and digital twin

In this section, we present the conceptual models for characterizing PTs and DTs.

4.1 Physical twin

PTs, i.e., CPSs in our context, appear in many diverse domains, such as communication, energy, healthcare, robotics, and transportation, as shown in enumeration *CPS Application Domain* in Figure 2. This classification was borrowed from the concept map of CPSs developed by Lee et al. [2]. Also, based on the concept map, we characterize a CPS with a list of *properties* such as being networked and/or distributed, being adaptive (responding to its changing *Physical Environment* and requests from its DT) and predictive (predicting changes in its *Physical Environment*), being intelligent in terms of self-learning, understanding and perception, and/or being real-time. Additionally, there are two other properties that are interleaved with these properties: uncertainty and evolution, which will be discussed separately in Section 5.

We would also like to acknowledge that there exist many conceptual models for characterizing CPSs in the literature. In this paper, we tried to characterize CPSs at the minimum level as readers can refer to many available descriptions

Table 1. Running Example for Conceptual Model – Overview

Concept(s)	**Explanation**
Physical Twin	An AWS and its constituent systems such as sorters, cranes, and conveyor belts.
Physical Process	A process to automatically create an order requested by a supermarket, including various quantities of various items such as breads and soda.
Physical Environment	Everything included in the operating environment of an AWS (including humans), characterized with factors such as the warehouse temperature.
DT and DT Environment	A digital replica of an AWS and its operating environment. Such a replica may consist of several simulators integrated with each other with the FMI standard [1] simulating the physical environment of the AWS.
DT Functionality	Examples include monitoring an AWS, raising alarms in case of emergencies, and predicting the time for the next warehouse maintenance.
DT-To-PT Connection/PT-to-DT Connection	Communication interfaces defined for transferring information from sensors of the AWS to its DT. The DT feeds back insights to the PT via actuators to intervene with the physical processes of the AWS.
Twinning	The DT is synchronized with its AWS every half an hour, for instance.
State	The current values of all the state variables of an AWS, its constituent systems, and environment.
State Change	The change in the current values of all the state variables of an AWS.
Parameter	Examples include values of the state variables of the sorter system in an AWS being transferred from the AWS and its DT.

Table 2. Running Example for Conceptual Model – Physical Twin

Concept(s)	**Explanation**
CPS Application Domain	The AWS CPS belongs to the manufacturing domain.
CPS Property	The AWS CPS is a networked, adaptive, and real-time system.

of CPSs at the conceptual level such as the concept map presented by Lee et al. [2] and the survey presented in [14].

The running case study illustrating this part of the conceptual model is shown in Table 2.

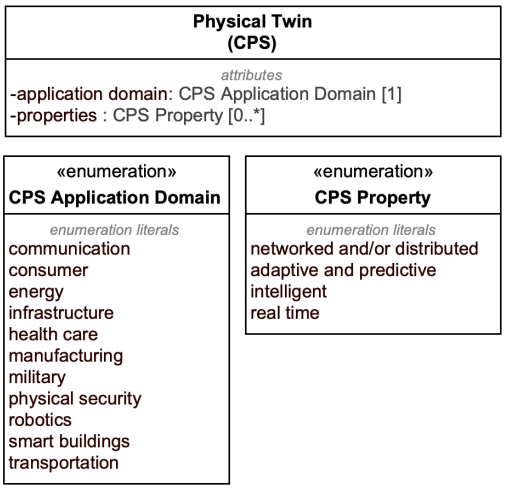

Fig. 2. Characterizing Cyber-Physical Systems (CPSs)

4.2 Digital twin

We first like to acknowledge that, in this section, we make the effort to characterize DTs from several aspects that we think are important. Therefore, it is, by no means, comprehensive. This part of the conceptual model is presented in Figure 3.

As shown in Figure 3, a *Digital Twin* can be operated by human operators, developed with specific DT platforms (e.g., GE Digital Twin[8]), supported by a diverse set of technologies such as machine learning and artificial intelligence (ML/AI), modeling and simulation techniques belonging to different paradigms (e.g., numerical modeling, software and system modeling, co-simulation supported with Simulink and SysML), as shown in the enumeration *Technology Type* of Figure 3.

A DT is situated in its *DT Environment* (as discussed in Section 3), which naturally requires a *Data Management Infrastructure*, as the DT needs to obtain data from the PT at real time, analyze the data received at real time and also historical data collected in the past, manage the data in terms of storing, sharing, authorizing the access of data, and so on.

The essential component of a DT is its *DT Model*, which aims to virtually represent a diverse set of aspects of its counterpart, i.e., the PT. As suggested by Solomon W. Golomb in [12] decades ago: 'Don't limit yourself to a single model, more than one may be useful for understanding different aspects of the same phenomenon'. Considering that engineering a CPS is inherently multi-disciplinary, a DT model, therefore, needs to be developed with modeling languages (e.g., SysML, Simulink, Modelica), methodologies and tools belonging

[8] `https://www.ge.com/digital/applications/digital-twin`

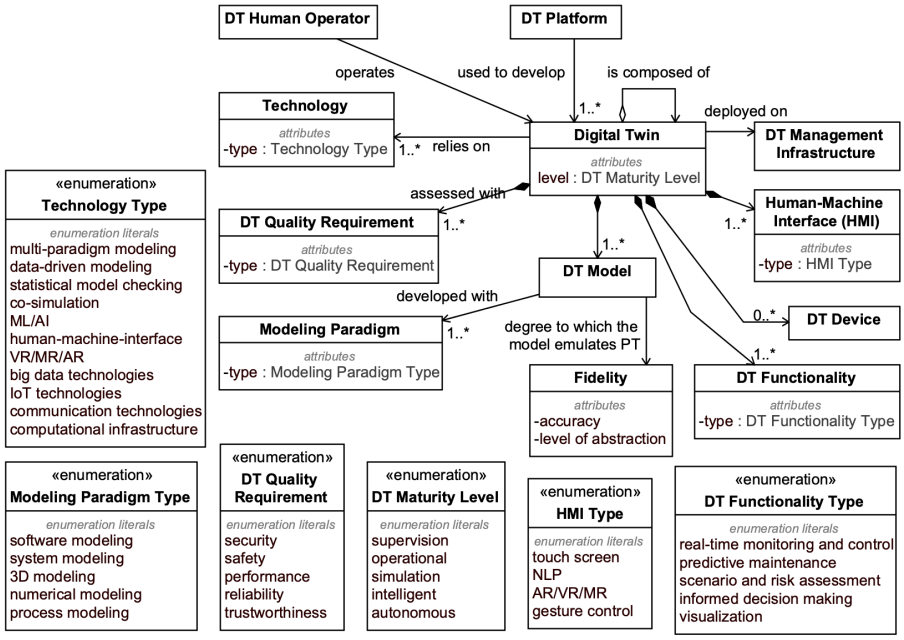

Fig. 3. Characterizing Digital Twins

to different modeling paradigms, some of which are shown in the enumeration *Modeling Paradigm Type* of Figure 3.

Fidelity is often used to characterize the degree to which a DT model imitates the PT. High fidelity models are often required to infer properties of real systems in the context of scientific modeling and simulations from the perspective of physics, for instance [18]. However, when a model already clearly serves its purpose in an engineering context, it is often unrealistic and even meaningless to pursue a higher fidelity, which often requires significant amount of effort and might result in low efficient models. Therefore, selecting suitable modeling paradigms and capturing different modeling aspects at suitable levels of abstractions will lead to multi-fidelity DT models, as also discussed in [5].

A DT is intended for achieving one or more specific functionalities such as real-time monitoring and control, predictive maintenance, scenario and risk assessment (e.g., unknown or uncertain scenarios, what-if analyses), efficient and informed decision making, and data/information visualization, as shown in the enumeration *DT Functionality Type* in Figure 3. Achieving these functionalities requires the support of various technologies (shown in enumeration *Technology Type*) and the achieved functionalities determine the maturity level of a DT. We borrow the different DT maturity levels defined in [26] and present them as enumeration *DT Maturity Level* in Figure 3. The first two levels offer *supervision*

via monitoring and manual control of the *operation* of the PT, respectively. At the third level, the DT can *simulate* the PT's behavior to provide a stronger decision support for the design and operation of the PT. Such decision support is further enhanced by benefiting from machine learning (ML) and advanced artificial intelligence (AI) techniques to automate decision support with minimal human intervention, to achieve the *intelligent* and *autonomous* maturity levels.

Same as for other software systems, a developed DT should be verified against a list of quality requirements such as security, safety, performance, reliability, and trustworthiness, as shown in the enumeration *DT Quality Requirement* in Figure 3. For example, as shown in Figure 1, a DT is bidirectionally connected with its corresponding PT to support *Twinning*, i.e., transferring data of parameters between twins at runtime through the established connections. Therefore, in certain application contexts (e.g., healthcare), a particular care should be taken to ensure no leaking of sensitive information (e.g., patient personal medical profiles).

A DT may be composed of a set of other DTs of the same or different types (*DT Aggregation* in Figure 3). This is because a complex CPS might be composed of different components, which are developed or operated with their own DTs. For example, an AWS might have a set of aggregated DTs corresponding to various subsystems (e.g., automated material handling systems, warehouse management and execution systems) for dealing with different physical processes. A better example is presented in [9], where two connected digital twins are suggested for the trauma management: managing the physical process of the pre-hospital phase with one DT, and the physical process of managing the trauma inside the hospital with another DT, as reported in [9].

A DT often needs a *Human-Machine Interface (HMI)*, which can be of various types and developed with different technologies such as Augmented Reality (AR)/Virtual Reality (VR), Natural Language Processing (NLP) enabled communication between humans and machine via voice, hand gesture-based control, or simply a control panel with or without a touch screen, as represented in enumeration *HMI Type* in Figure 3. HMI is important for DTs because being fully autonomous (without human intervention, having the *DT Maturity Level* being the highest, i.e., *autonomous*) is not realistic in a lot of application contexts; therefore, operating DTs will be highly dependent on human interactions and, consequently, their designs need to take care of the human-in-the-loop aspect by selecting and applying appropriate HMI techniques. Dependent on the design of HMI, additional devices might need to be introduced to DTs, represented as concept *DT Device* in Figure 3. A relevant work is presented in [13], where the authors proposed to enhance the capability of DTs with VR in the context of manufacturing Cyber-Physical Production System.

The running case study illustrating this part of the conceptual model is shown in Table 3.

Table 3. Running Example for Conceptual Model – Digital Twin

Concept(s)	**Explanation**
Human-Machine Interface	The HMI of the DT of an AWS can come in different shapes, for instance, designed to have a touch screen.
Data Management Infrastructure	The DT can use various data management infrastructures which can be cloud-based solutions or in-house data centers.
DT Aggregation	It is realistic to develop a DT that is an aggregation of a set of DTs, which corresponds to various subsystems of an AWS such as automated material handling systems and warehouse management and execution systems.
DT Platform	Existing platforms for developing DTs can be used.
DT Human Operator	DT operators monitor the operation of the AWS through the provided HMI, identify potential improvements to the AWS operation, and intervene with its operation when needed, with the help of data analytic of the DT.
Technology	Different sensors attached to monitor the surface of conveyor belts in an AWS. Such sensors help to determine when to clean the surface of the conveyor belts.
DT Quality Requirement	The DT and the AWS are connected through internet; therefore, any unauthorized access to the DT should be prevented.
DT Maturity Level	The DT of the AWS provides supervision facility and allows running simulations to assess the performance of the operational AWS.
DT Model and Modeling Paradigm	The DT of an AWS can consist of various hardware models of various systems (e.g., sorters), in Simulink for instance. In addition, the environment of the DT of the AWS can be modeled with numerical models. Moreover, the warehouse management process can be modeled with Petri Nets.
Fidelity	Depending on what functionalities a DT implements, the DT model might have different fidelity levels. For instance, 3D model of high fidelity is often used to allow users to visualize certain pallets in the warehouse and help operators to have an overview of the warehouse such that the operators might identify opportunities to improve stock placement and picking processes. Such a high fidelity model is however inappropriate for visualizing goods quantities, and data visualization techniques such as customizable charts and tables might better serve the purpose.

5 Evolution and uncertainty

As previously discussed, we consider evolution and uncertainty as two important aspects to be considered when developing, operating and maintaining DTs and CPSs. Therefore, in this section, we provide our vision and call for contributions on these two aspects.

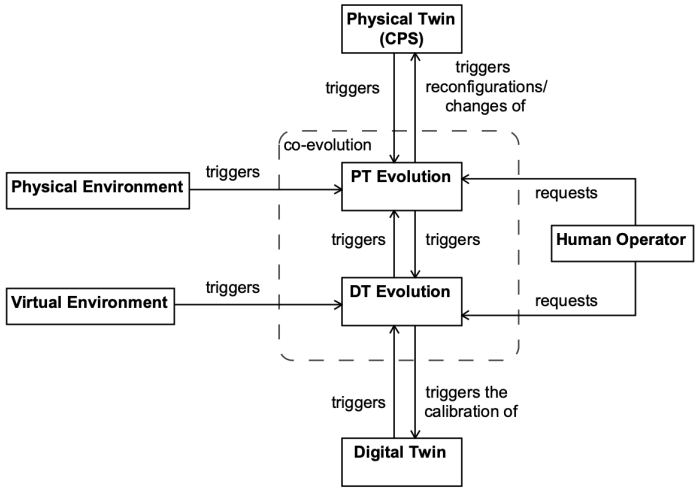

Fig. 4. PT and DT Evolution

5.1 Evolution

An intelligent PT naturally evolves during its operation by constantly interacting with its physical environment (probably partially known at a given time point), humans and its DT, which we call *PT Evolution* (Figure 4). Consequently, a DT also naturally evolves when it interacts with the virtual environment (probably presenting a partial view of the physical environment as this is partially known at a given time point), and continuously receives data from its PT, i.e., *DT Evolution*. As shown in Figure 4, *PT Evolution* triggers *DT Evolution*, and vice versa.

Specifically, *DT Evolution* is often triggered in the following situations:

- The PT evolves as a result of a software update or replacement of a hardware component, etc., which consequently triggers the evolution of the DT;
- The DT needs to be evolved to accommodate a new *DT Functionality*;
- The DT needs to be evolved to refine the *DT Model* when more data from the PT becomes available and more information becomes available during its interactions with the virtual environment and human operators via HMI;
- An adaptive and autonomous DT self-evolves when its operating environment changes, through its implemented self-configuration mechanism for instance.

PT Evolution is often triggered in three situations:

- The upgrade or replacement of its own software and hardware components;
- Configurations from human operators or the DT received during its operation and maintenance;

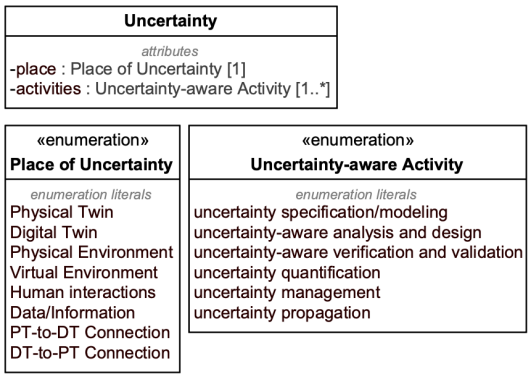

Fig. 5. Characterizing Uncertainty

– Self-adaptation or self-configuration implemented in an adaptive and/or autonomous physical twin.

Therefore, this situation is an instance of *co-evolution* of two intelligent systems. Particular designs of the twins themselves and the twinning between them are needed to cost-effectively accommodate such co-evolution, which, to the best of our knowledge, has not received sufficient attention probably because applying DTs to empower CPSs is still at its early stage.

5.2 Uncertainty

In the last decade, uncertainty, especially external uncertainty in the open and operational environment in CPSs (e.g., [29], [3]) and self-adaptive systems (e.g., [19], [20]), has attracted a lot of attention because uncertainty is an unavoidable feature of such systems. Properly dealing with uncertainty in DTs and CPSs is especially critical, considering the increasing complexity in terms of the scales of the digital and physical twins, network communications within and between the twins, and/or deployed ML/AI algorithms in both of the twins, and their ever-changing and open operating environment. Therefore, as presented in enumeration *Place of Uncertainty* in Figure 5, different places of the digital and physical twins, the twinning between them, their respective environments, interactions with humans, and even data/information transferred collected by the twins and transferred across the twins, all possibly contain uncertain information. Therefore, uncertainty is yet another dimension that significantly increases the complexity of engineering both DTs and CPSs.

There exist related works on specifying, modeling and measuring/quantifying uncertainty in the context of CPSs. For example, Zhang et al. [30] proposed a conceptual model (named *U-Model*) to understand and characterize uncertainty and its associated concepts from the angle of software engineering and especially model-based engineering. Later on, Zhang et al. also instantiated U-Model for

Table 4. Running Example for Conceptual Model – Evolution & Uncertainty

Concept(s)	**Explanation**
PT Evolution	The introduction of a new sorter hardware to the AWS.
DT Evolution	New and high quality sensors are added to monitor the surface of the sorter systems.
Uncertainty	The AWS system sometimes stops without any obvious reason, which requires manual restart of the AWS. This is an example of uncertainty in the PT. The network connection between the DT and PT might get lost occasionally, which should be considered as a kind of uncertainty in the connections between the DT and the PT. There might be uncertainty in DT-recommended warehouse maintenance strategies.

specifying system requirements as use case models [31] and for modeling uncertainty by extending standard UML state machine notations [29]. Along the same line, in the context of dynamically adaptive systems, Betty et al. [8] proposed RELAX [8] to support uncertainty specification and analysis. Later on, FLAGS [4] was proposed for enabling the specification of adaptive goals. To support decision making under uncertainty, various methodologies have been proposed for testing CPSs under uncertainty [28,7], etc.

Though these related works have built the foundation for developing uncertainty-aware solutions in the context of engineering DTs for CPSs, it lacks systematic solutions going from uncertainty specification, modeling and analysis, design and development of solutions to support decision making under uncertainty, all the way down to uncertainty-wise verification and validation, as shown in enumeration *Uncertainty-aware Activity* in Figure 5. Especially during its operation, the DT constantly receives data from its corresponding PT, which can be used to discover (previously) unknown information from these received data and, consequently, make more informed decisions. It is expected that specific DT functionalities are hence needed to elegantly deal with uncertainties that may exist or occur in different places (enumeration *Place of Uncertainty* in Figure 5).

An instantiation of the running case study for these conceptual models is shown in Table 4.

6 Life-cycle

A DT co-exists with its counterpart and it is expected to support the full life-cycle of its PT. As a complex system itself, a DT has its own development life-cycle. As shown in Figure 6, inspired by the concept model developed by Lee et al. [2], we consider that both the life-cycles of a DT and its PT are composed of a list of essential activities, captured in enumeration *Essential Activity Type*. We acknowledge that there exist standards and methodologies (e.g., ISO/IEC/IEEE 15288) for developing CPSs, but we are not aware of any standard or methodology for developing DTs.

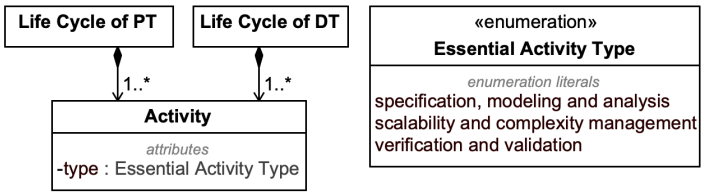

Fig. 6. DT and PT Life-cycles and Essential Activities

In addition, to support a CPS development and operation through a DT, a fundamental research question is: how to coordinate the life-cycle of the DT with the life-cycle of its counterpart CPS? Below, we list three such DT and CPS dual life-cycle options (with in total five different settings):

Option 1: CPS and DT do not exist. This is the typical case when a very complex and safety-critical CPS (e.g., an entirely new smart hospital) need to be built, and the usage of a DT is chosen to strengthen, for instance, the safety and the performance of the CPS. There are three settings for this option: 1) Building a CPS and its DT together right from scratch; 2) Starting with building a CPS, and, at a later point in time, starting the development of the DT and aligning its development with its CPS); 3) Starting with building a DT for a CPS, and later introducing the development of the CPS itself, with the DT that has been (partially) developed as the basis.

Option 2: DT exists. It starts developing a CPS from an existing DT. This is the typical case that can occur in product lines, i.e., a DT DT_1 has been developed in the past for a product p_1, and the DT DT_2 for another new product p_2 is derived by extending DT_1 (reasoning on the similarities and differences between p_1 and p_2).

Option 3: CPS exists. It builds a DT for an operational CPS to better support its operation and maintenance. This case happens when the CPS has been developed in the past, but further requirements on its operation have emerged. If its re-engineering is not feasible, the development of a DT can be a cost-effective solution.

Rigorous methodologies are needed to guide the development of a DT and coordinate with the development life-cycle of the counterpart CPS. Also, the *Twining Process* (see Figure 1) brings another layer of complexity to the *dual* development life-cycles of the twins. We also acknowledge that developing the twins is naturally a highly iterative process due to their complex nature.

The running case study illustrating this part of the conceptual model is shown in Table 5.

7 Related work

Jones et al. [16] presented a systematic literature review of 92 digital twin publications and identified a list of concepts such as physical entity, virtual entity,

Table 5. Running Example for Conceptual Model – Life-cycle

Concept(s)	**Explanation**
Life-cycle of PT	The AWS system is in the operational phase.
Life-cycle of DT	The DT is in the operational phase.
Activity	The development of an AWS and its corresponding DT often requires to go through some of the phases defined in enumeration *Essential Activity Type* shown in Figure 6.

fidelity, and twining. In our paper, we constructed the conceptual model presented in Figure 1 based on the review results of this paper. The authors of the paper also presented a list of future directions and research gaps. For example, they discovered that the majority of the works studies have put their focuses on the implementation phase of the DT life-cycle and insufficient attention has been given to the early phases of the DT life-cycle. In addition, the majority of identified use cases of DTs are manufacturing-related, and talk about simulation modeling, optimization, and data management. Similarly, another literature review was presented by Josifovska et al. in [17], which defines concepts similar to those of Jones et al. [16].

Negri et al. [21] conducted a literature review of 26 publications to answer two questions: how DTs are defined in the literature and what is the role of a DT in Industry 4.0. The authors concluded that: (1) the scientific literature is still immature as the studied literature mostly refers to different definitions of DTs; and (2) in the manufacturing industry, it is relevant to define a DT as a virtual counterpart (a digital representation) of a physical device, which is augmented/updated/synchronized with data continuously from the physical object, to support decision making and predictive maintenance.

Tao et al. [24] presented their vision of engineering DTs for CPSs of the manufacturing domain, by providing a mapping between the physical and digital worlds, and a hierarchical structure that divides CPSs and DTs into three levels: the unit, system, and system of systems (SoS) levels. In this structure, DTs are connected to their corresponding CPSs at every single level. For instance, a unit-level DT is actually a model of a physical object specifying its geometric shape, identity, operating status, and even its high-fidelity visual simulation. A system-level DT is an integration of the models constructed at the unit level. In addition, the system-level CPS and DT share the same architecture. For instance, an aircraft is composed of various components (e.g., engines, wings), each of which has a DT. An SoS-level DT is simply the integration of the system-level DTs. This structure naturally implies the development life-cycle of going from the unit-level to the SoS-level, which is the same for both CPSs and DTs. The authors also explicitly say that they consider CPSs and DTs as conceptually similar in smart manufacturing, and the key difference is that DTs are more concerned about models and data.

Rasheed et al. [22] also presented a literature review of methodologies and techniques for constructing DTs with the aim of identifying values of applying DTs (e.g., supporting what-if analysis and informed decision making), their application domains (e.g., health, meteorology, and manufacturing), and current challenges (e.g., large-scale data fusion, data security, and real-time simulations). In addition, the paper also discussed enabling technologies such as physics-based modeling, data-driven modeling, big data and IoT technologies, and human-machine interface.

Xiaodong et al. [27] reviewed the use of DTs for Prognostics and Health Management (PHM) in the literature. They identified that, in this context, the main functionalities of a DT are (i) keeping historical data about the PT, (ii) monitoring its health and performing fault diagnosis, (iii) devising an efficient maintenance schedule for the PT, and (iv) allowing testing in a virtual environment to avoid damages to the PT occurring from destructive tests. The concepts are illustrated with a high-speed train electric multiple unit.

Boschert and Rosen [5] focused on the simulation aspects of the DT. Apart from common concepts also presented in other papers, the authors underline the importance of selecting the suitable abstraction of the model for the problem to be solved, as having unnecessarily too-detailed models can lead to scalability issues. Moreover, they underline that the simulation facilities of the DT make it suitable to be used along all the life-cycle of the physical twin, not only in operation. In the design phase, for example, the models that are created for building the system are a first foundation of the DT itself.

8 Conclusion

Digital twins (DTs) for Cyber-Physical Systems (CPSs) aim to improve the current practice of developing and operating CPSs. The increased interest in DTs for CPSs have resulted in various definitions of DTs and their associated concepts. Thus, in this paper, we presented a conceptual model of different DTs concepts and their relationships based on the published literature. Moreover, concepts are also explained with a running example. The conceptual model serves as the first step towards providing a unified meaning of various DT concepts in the CPS community. In addition, we acknowledge the needs of developing systematic solutions for dealing various uncertainties in the development and operation of both CPSs and their DTs, and their co-evolution. We also encourage more research activities on designing dual lifecycles of engineering DTs and CPSs.

References

1. The FMI Standard. `https://fmi-standard.org/`
2. Asare, P., Bernard, R., Broman, D., Lee, E.A., Prinsloo, G., Torngren, M., Sunder, S.S.: Cyber-Physical Systems - a Concept Map. `http://cyberphysicalsystems.org/`

3. Bandyszak, T., Daun, M., Tenbergen, B., Weyer, T.: Model-based documentation of context uncertainty for cyber-physical systems. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). pp. 1087–1092. IEEE (2018)
4. Baresi, L., Pasquale, L., Spoletini, P.: Fuzzy goals for requirements-driven adaptation. In: 2010 18th IEEE International Requirements Engineering Conference. pp. 125–134. IEEE (2010)
5. Boschert, S., Rosen, R.: Digital Twin—The Simulation Aspect, pp. 59–74. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-32156-1_5
6. Cameron, D.B., Waaler, A., Komulainen, T.M.: Oil and gas digital twins after twenty years. how can they be made sustainable, maintainable and useful? In: Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), 26-28 September 2018, Oslo Metropolitan University, Norway. pp. 9–16. Linköping University Electronic Press (2018)
7. Camilli, M., Bellettini, C., Gargantini, A., Scandurra, P.: Online model-based testing under uncertainty. In: 2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE). pp. 36–46. IEEE (2018)
8. Cheng, B.H.C., Sawyer, P., Bencomo, N., Whittle, J.: A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. In: Model Driven Engineering Languages and Systems. pp. 468–483. Springer Berlin Heidelberg (2009)
9. Croatti, A., Gabellini, M., Montagna, S., Ricci, A.: On the integration of agents and digital twins in healthcare. Journal of Medical Systems **44**(9), 1–8 (2020)
10. Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., Yamu, C.: Urban digital twins for smart cities and citizens: The case study of herrenberg, germany. Sustainability **12**(6), 2307 (Mar 2020). https://doi.org/10.3390/su12062307
11. Fonseca, Í.A., Gaspar, H.M.: Challenges when creating a cohesive digital twin ship: a data modelling perspective. Ship Technology Research pp. 1–14 (2020)
12. Golomb, S.W.: Mathematical models: Uses and limitations. IEEE Transactions on Reliability **R-20**(3), 130–131 (1971)
13. Havard, V., Jeanne, B., Lacomblez, M., Baudry, D.: Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations. Production & Manufacturing Research **7**(1), 472–489 (2019)
14. Hehenberger, P., Vogel-Heuser, B., Bradley, D., Eynard, B., Tomiyama, T., Achiche, S.: Design, modelling, simulation and integration of cyber physical systems: Methods and applications. Computers in Industry **82**, 273–289 (oct 2016). https://doi.org/10.1016/j.compind.2016.05.006
15. Iglesias, A., Lu, H., Arellano, C., Yue, T., Ali, S., Sagardui, G.: Product line engineering of monitoring functionality in industrial cyber-physical systems: A domain analysis. In: Proceedings of the 21st International Systems and Software Product Line Conference - Volume A. p. 195–204. SPLC '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3106195.3106223
16. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology (2020). https://doi.org/10.1016/j.cirpj.2020.02.002
17. Josifovska, K., Yigitbas, E., Engels, G.: Reference framework for digital twins within cyber-physical systems. In: Proceedings of the 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems. p. 25–31. SEsCPS '19, IEEE Press (2019). https://doi.org/10.1109/SEsCPS.2019.00012

18. Lee, E.: The past, present and future of cyber-physical systems: A focus on models. Sensors (Basel, Switzerland) **15**, 4837–4869 (03 2015). https://doi.org/10.3390/s150304837
19. Ma, T., Ali, S., Yue, T., Elaasar, M.: Testing self-healing cyber-physical systems under uncertainty: a fragility-oriented approach. Software Quality Journal **27**(2), 615–649 (2019)
20. Moreno, G.A., Cámara, J., Garlan, D., Klein, M.: Uncertainty reduction in self-adaptive systems. In: Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems. pp. 51–57 (2018)
21. Negri, E., Fumagalli, L., Macchi, M.: A review of the roles of digital twin in cps-based production systems. Procedia Manufacturing **11**, 939 – 948 (2017). https://doi.org/10.1016/j.promfg.2017.07.198
22. Rasheed, A., San, O., Kvamsdal, T.: Digital twin: Values, challenges and enablers from a modeling perspective. IEEE Access **8**, 21980–22012 (2020)
23. Safdar, S.A., Yue, T., Ali, S., Lu, H.: Evaluating variability modeling techniques for supporting cyber-physical system product line engineering. In: Grabowski, J., Herbold, S. (eds.) System Analysis and Modeling. Technology-Specific Aspects of Models. pp. 1–19. Springer International Publishing, Cham (2016)
24. Tao, F., Qi, Q., Wang, L., Nee, A.: Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. Engineering **5**(4), 653–661 (2019). https://doi.org/10.1016/j.eng.2019.01.014
25. Tao, F., Zhang, H., Liu, A., Nee, A.Y.: Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics **15**(4), 2405–2415 (2018)
26. Wagg, D.: Asset Management using the Digital Twin concept. `https://www.thefuturefactory.com/blog/26`, last accessed: September 4, 2020
27. Xiaodong, W., Feng, L., Junhua, R., Rongyu, L.: A survey of digital twin technology for PHM. In: Jain, V., Patnaik, S., Popențiu Vlădicescu, F., Sethi, I.K. (eds.) Recent Trends in Intelligent Computing, Communication and Devices. pp. 397–403. Springer Singapore, Singapore (2020)
28. Zhang, M., Ali, S., Yue, T.: Uncertainty-wise test case generation and minimization for cyber-physical systems. Journal of Systems and Software **153**, 1 – 21 (2019). https://doi.org/10.1016/j.jss.2019.03.011
29. Zhang, M., Ali, S., Yue, T., Norgren, R., Okariz, O.: Uncertainty-wise cyber-physical system test modeling. Softw. Syst. Model. **18**(2), 1379–1418 (Apr 2019). https://doi.org/10.1007/s10270-017-0609-6
30. Zhang, M., Selic, B., Ali, S., Yue, T., Okariz, O., Norgren, R.: Understanding uncertainty in cyber-physical systems: A conceptual model. In: Wąsowski, A., Lönn, H. (eds.) Modelling Foundations and Applications. pp. 247–264. Springer International Publishing, Cham (2016)
31. Zhang, M., Yue, T., Ali, S., Selic, B., Okariz, O., Norgre, R., Intxausti, K.: Specifying uncertainty in use case models. Journal of Systems and Software **144**, 573–603 (2018)