CONTRIBUTED PAPER

Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models

This survey covers state-of-the-art approaches and discusses major technologies that have contributed to the field of Digital Twin.

By Roberto Minerva, *Senior Member IEEE*, Gyu Myoung Lee, *Senior Member IEEE*, and Noël Crespi, *Senior Member IEEE*

ABSTRACT | Digital twin (DT) is an emerging concept that is gaining attention in various industries. It refers to the ability to clone a physical object (PO) into a software counterpart. The softwarized object, termed logical object, reflects all the important properties and characteristics of the original object within a specific application context. To fully determine the expected properties of the DT, this article surveys the state-of-the-art starting from the original definition within the manufacturing industry. It takes into account related proposals emerging in other fields, namely augmented and virtual reality (e.g., avatars), multiagent systems, and virtualization. This survey thereby allows for the identification of an extensive set of DT features that point to the "softwarization" of POs. To properly consolidate a shared DT definition, a set of foundational properties is identified and proposed as a common ground outlining the essential characteristics (must-haves) of a DT. Once the DT definition has been consolidated, its technical and business value is discussed in terms of applicability and opportunities. Four application scenarios illustrate how the DT concept can be used and how some industries are applying it. The scenarios also lead to a generic DT architectural model. This analysis is then complemented by the identification of software architecture models and guidelines in order to present a general functional framework for the DT. This article, eventually, analyses a set of possible evolution paths for the DT considering its possible usage as a major enabler for the softwarization process.

KEYWORDS | Artificial intelligence (AI); business models; cyber physical systems (CPSs); digital twin (**DT**); Internet of Things (IoT); machine learning (ML); multiagent systems; network function virtualization; sensors; servitization; smart city; software architecture; softwarization; virtual and augmented reality.

I. INTRODUCTION

The digital twin (DT) concept has been attracting increasing attention for its ability to create a software counterpart of a physical object (PO). The DT concept was originally conceived by Michael Grieves and presented in 2003 at the University of Michigan [1]. Since then, the DT model has attracted significant interests, both in academia and industry. Usage of DT first grew in the manufacturing environment and later in the community of the Internet of Things (IoT) and cyber physical systems (CPSs). It has also drawn the interest of other technical communities and of practitioners in several industries. They have found communalities with their own approaches, ideas, and requirements. In this way, the DT concept has been applied and extended to the point where different facets can be assumed, depending on the application domain and the intended usage. One important point is that the definition has moved from an industrial artifact or product to a more generic notion applicable to almost any PO and, in principle, to intangible objects.

As a starting definition, strongly based on [2], this article will initially adopt the following statement describing

Manuscript received February 29, 2020; revised May 18, 2020; accepted May 24, 2020. Date of publication June 18, 2020; date of current version September 25, 2020. *(Corresponding author: Roberto Minerva.)*

Roberto Minerva and **Noël Crespi** are with CNRS Lab UMR5157, Telecom SudParis, Institut Polytechnique de Paris, Evry, France (e-mail: roberto.minerva@telecom-sudparis.eu; noel.crespi@telecom-sudparis.eu).

Gyu Myoung Lee is with the Computer Science Department, Liverpool John Moores University, Liverpool L3 3AF, U.K. (e-mail: g.m.lee@ljmu.ac.uk).

Digital Object Identifier 10.1109/JPROC.2020.2998530

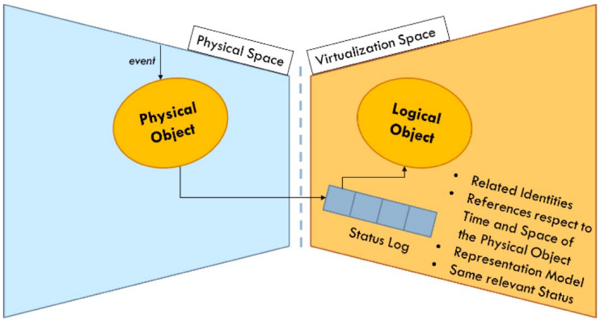

Fig. 1. ***Representation of the DT.***

the DT: "a DT is a comprehensive software representation of an individual PO. It includes the properties, conditions, and behavior(s) of the real-life object through models and data. A DT is a set of realistic models that can simulate an object's behavior in the deployed environment. The DT represents and reflects its physical twin and remains its virtual counterpart across the object's entire lifecycle." This definition, represented in Fig. 1, has been adjusted by substituting "product" with " PO" and "digital representation" with "software representation." This change is due to making it more general and applicable also in other contexts, for example, IoT and CPS. From the incipit, the DT has gone through evolution and progression. Several explicit or implicit extensions to the concept have been added depending on specific problem domains. A consolidated DT (cDT) definition will be devised over the following three chapters. A cDT defines the general concepts by means of well identified and shared properties. It also encompasses mechanisms and functions from different technological areas. This consolidation is needed in order to generalize the definition, and then the implementation, and exploitation of relevant properties in several problem domains.

The DT concept is simple enough to be understood and potentially applied in several contexts. It has attracted more and more attention in several different areas. At the same time, it encompasses conceptions, properties, and expected functionalities from different technological and application areas. One of the goals of this article is to identify and emphasize the key features of the DT concept in order to make it a general concept applicable in the IoT realm. An extensive literature has been produced about DT, but, sometimes, different words or different facets of the DT have been considered, defined, and stressed. There is a need to clarify the major common features and to bring clarity to existing interpretations in order to propose a unifying framework for the DT. This article highlights the various intertwining views and the expectations around the concept in IoT and related fields of information and communication technologies (ICTs) domain. This article also provides a general outline of the intended properties and characteristics that have been defined and that overlapped over time. In order to credit and recognize the different contributions and requirements emerging from different technological areas, this article introduces a functionally consolidated definition of a DT beyond the concept originally developed for product lifecycle management and manufacturing in general.

This article is a survey of major definitions, specifications, and implementations of the DT concept in several technological areas, and it is also an attempt to consolidate the major features of the DT concept as it has emerged in different industries. This article also aims to survey its applicability in relevant IoT application scenarios and to prepare a framework for assessing the possibility of its implementation in modern software architectures. This article mainly emphasizes the software side of the concept and its relationship to middleware architectures for the IoT [3]. The DT is, in fact, having a double role in IoT, on the one side it is implemented and recognized as a major approach for creating IoT applications; on the other side, the DT is naturally associated with the ability of sensing and actuation of IoT technologies [4]. Hence, DT implementations have deep relationship with IoT capabilities.

A set of general questions that this article raises for investigation include the following.

1) What is a DT in a software context? What are the supporting and available technologies? How is the DT concept being used and exploited? These topics will be presented and discussed in Section II.
2) Is there a shared/common DT definition? Are there properties that can fully characterize the DT definition? These topics will be dealt with in Section III.
3) What are the essential features and properties of a DT-based architecture? Is the concept of DT adding value to software architectures? And in particular IoT architectures? Is there an ecosystem, or is it possible to build one, that is taking or that could take, advantage of it? Are there missing parts that hamper the full exploitation of the DT concept's capabilities? What are some of the academic and industrial platforms that can help to enforce and cDT solutions? The reader will find these topics in Section IV.
4) What are some of the ways the DT could be utilized in an IoT context? What are some of the important scenarios that could actually benefit from the usage of DT's functions and features? These matters are analyzed in Section V.
5) What is the possible evolution of the concept? What is its real applicability in actual contexts? What are the emerging architectural models? These themes are presented in Section VI.

In order to identify a common characterization of DT concept and to provide some answers to the questions at hand, this article is organized as follows: Section II is a survey of the state of the art. It discusses the mainstream technologies and trends that have produced or directly contributed tools or technologies useful for the DT definition and understanding. It also considers approaches that have been inspired by the DT. Section III focuses on basic features of the DT derived from the developments discussed in the state of the art. A "consolidated" definition of the DT concept, with respect to different technological paths' contributions, is provided in terms of foundational aspects. This part can be seen as a mini survey in itself, devoted to the understanding of the real-life properties and important features of the DT concept. The value of this cDT is then discussed from technical and business perspectives in Section IV. Here, some approaches and possibilities that give a real value to the DT implementation and exploitation are identified, for example, interoperability. Section IV also offers suggestions and guidelines for determining the real value of the application and implementation of the concept and what actors could benefit. Section V then presents some of the many applications of the DT concept in IoT context. Providing a survey on the applicability of the DT concept in several interesting contexts is another goal of this section. In order to guide the user through the complexity of possible developments, this article describes how specific applications can be built, referring to activities and experiences conducted in similar or otherwise meaningful experiments. This section offers some guidelines on how to apply the DT to satisfy the requirements of a specific scenario in order to introduce applications of increasing complexity.

Section VI analyzes some actual architectures supporting the DT concept and seeks to relate them to the identified properties. It also relates them to a possible general reference framework of the cDT. This section can be seen as a survey of some academic and industrial platforms, with simple descriptions on how to exploit the platforms' capabilities to realize a fully fledged DT solution.

Finally, Section VII discusses the issues, problems, or obstacles that should be considered, and which evolution scenarios are most probable or preferable for the DT application.

II. STATE OF THE ART

This section presents the current paradigms, assumptions, and models under which many researches have defined and applied the DT. This article focuses on how IoT systems can benefit from the DT concept and what types of uses can be realized. Different application domains are considered. This article specifies what types of basic definitions of the DT concept have been put forward, how the concept has been defined, and more importantly, applied, and what problems it has helped to solve. In addition, this section identifies and relates the different scientific and technological trends that have influenced the current understanding and definition of the DT concept. The different paths have different views and definitions of the DT. They refer to the general idea of replicating, by software, POs, or products. This is the level at which the DT definitions are "conceptually" aligned, but they differ in terms of properties and emphasis on specific characteristics. Finding a set of characterizing properties of the DT within different application domains is then fundamental in order to create a common ground for the definition and the application of the concept.

A. Contributions From Manufacturing Studies

The concept of the DT [5] originated in the manufacturing domain. It is a concept that has been especially helpful in guiding a deep theoretical and practical change in how products are designed, realized, used, and disposed. A DT is intended to span the entire lifecycle of a product, enabling the design, prototyping, testing, production, and use of a virtual representation of a product, that is, a PO. The physical and digital/virtual counterparts are explicitly related to each other and thus can be used to fully design, experiment, understand, and measure the physical characteristics of the real object at any stage of the product's lifecycle.

Usually products are defined and developed to be used in complex environments. The DT concept is also considered as a means to cope with the emergent behavior of complex systems [6]. A DT should be able to represent and act like a real object even in large systems whose behavior could change over time according to changing conditions. It is important to note that Grieves' definition of DT [6] does not apply to systems that show an evolutionary emergence, that is, systems that exhibit a deliberate capability to learn and modify their behavior in order to adapt to changing conditions. On the other side, there is a current trend in IoT and Industrial IoT (IIoT) toward the ability to implement adaptive systems [7], [8]. There is an increasing need to develop products or to control processes in an adaptive manner.

In manufacturing, a DT can be used to fully specify the product and to understand its inherent characteristics, features, and behaviors. According to Grieves and Vickers [6], some behaviors are purposely intended and designed during the definition of a product, while other behaviors, characteristics, and effects are not considered during the design, testing, or usage phases. These "unpredicted" characteristics could be positive or quite negative being effects of design or project mistakes. The DT can be used to determine all the unwanted and unexpected behaviors very early in the product lifecycle and thus to help to correct them. This early detection is possible because the virtual representation of a product can be prototyped and "tested" in many more situations and conditions than traditional physical prototypes. Given this perspective, Grieves provides a definition: "the Digital Twin is a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level" and as a corollary he explains that: "at its optimum, any information that could be obtained from inspecting a physical manufactured product can be obtained from its

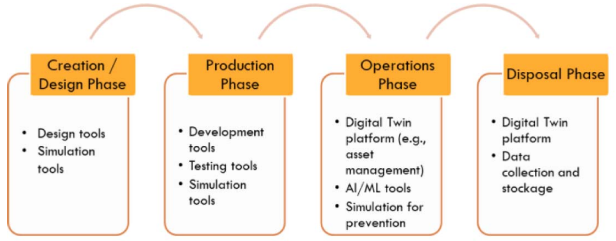

Fig. 2. ***Lifecycle of a DT and some needed functions/tools.***

DT." A first important set of features of the definition of the DTs must be emphasized.

1) A DT strictly refers to a PO.
2) A DT contains all the information needed to fully characterize a PO and its intended or predicted behavior.
3) Since the DT is framed in a lifecycle composed of different steps, it can encompass data and information that describe the "history" of the PO.

The DT has an impact also on the actual management of the product life cycle, process lifecycle management (PLM) [6]. A simplified view of the PLM takes into consideration the creation/design phase in which the product is conceived and designed; the production phase during which the product is actually manufactured and realized; the operations phase during which the product is operated and actually used; and the disposal phase when the product is taken out of production, operation, and eventually dismissed. Two concepts are important from the manufacturing perspective: the life cycle and the software implementation of the DT. The DT finds its usage and utility in each of the different phases of the life cycle and the software counterpart helps in improving and optimize the "product" at each step. Fig. 2 represents a simplified DT life cycle as well as some supporting tools and functionalities to exploit the approach.

In each phase, tools, and functions are needed in order to properly execute the processes. Some tools are listed in different phases. This is because they are used for different purposes. For example, simulation in the creation phase may be used to choose some product options; in the production phase it may be used to simulate some expected behavior of the product, while in the operations phase, it may be used to check and predict some malfunctioning if the product is "stressed" or used in critical situations.

During the design phase, the DT will be represented by a logical object (LO), actually the only existing object that is a software archetype of all the POs to come. Once the product and its digital counterpart are out of the design phase, the production phase relates prototypes, and their software representation in order to test and experiment with the future product. In this phase, the software aspects of the DT help in optimizing the PO and in carrying out tests that otherwise would require the implementation of mockups. In the operation phase, the relationship between the LO and the products, that is, the POs, can be instantiated in different ways: a 1:1 definition means that one PO is represented by a single LO, while an 1: N refers to the fact that n physical implementations of product refer to one LO. In other words, a DT relates n products to a single software representation. In the latter case, several physical copies can refer to an archetypical object and cooperate in order to represent the actual capabilities of the class of instantiated objects. In some sense, the DT could be a metasystem representing the typical behavior of any of the physical instances of the product. An evident role for IoT technologies and capabilities is in production and operation phases. During these phases, sensors, and IoT platforms can be used to actually building products and later on in sensing and measuring the behavior and performance of products.

Another characterizing feature of the DT concept is the "linkage" between LO and PO. In the design phase, the features, data, information, and the model of the LO are obviously predominating on those of the PO, which may even not exist yet or be only a simple mockup. From the production phase onward, the information about or from the PO(s) must be collected and provided to the LO. This flow of information creates a linkage between the PO and LO. Thanks to the increasing capabilities of communications, the possibility that the PO and LO are connected by means of the Internet or other specific networks can often be assumed. This is not necessarily always the case, as a DT could be fed data by uploading, for example, through a storage media, measures collected in the PO and then uploaded to the LO. The linkage is not necessarily real time, nor resilient, or permanent. The flow is from the actual object(s) to the logical one. The actual states, changes, and any reactions of the PO should somehow be represented as information within the LO. The "direction" of the information is mainly, if not exclusively, from the physical to the virtual. However, it could be useful to have the possibility to have the LO to send data and information to the POs. For example, during the operation phase, the flow could also be from the virtual to the physical in order to initialize systems or to correct some states or errors: reinitialization of a machine after a break or an outage; a synchronization of states with other cooperating robots; or simply the initialization of permission and personalization when a specific customer is using a vehicle. Another important characteristic of the DT is its continuous synchronization with the production system and its evolution, for example, changes in wiring, physical fixation position, etc. During its life cycle, a DT must be able to synchronize with new or update engineering models and processes during design, production, and operation phases [9], [10].

Another important feature of the DT to exploit in different phases is simulation [11], [12]. It can be used to simulate and predict the behavior of the PO in a particular system, situation, or environment. It could be used to anticipate and prevent issues and disruptions of the PO under simulated circumstances. In this case, the information exchange could be bidirectional with real data measures and events flowing from real to virtual, and predictions, change of states and possible commands flowing in the other direction.

This linkage between the PO and the LO is also a key element for introducing new capabilities and thus new business opportunities for the industry. If a robust and permanent link between the PO, owned by a customer, and the logical one is possible, then the product could be tailored to a specific customer in terms of specialized features and/or additional functionalities [13]. This permanent linkage could be used to guarantee the validity and originality of the product to the customer, and for the "servitization" of the product itself, that is, to sell a product by means of (paid) access to its services. The linkage is a relevant feature of the DT. It relates the PO and the LOs and allows their synchronization. From a practical perspective, the linkage is subject to the issue of distributed applications [14] and always-on devices. For instance, latency and reliability issues may introduce disruption in the DT. However, the recent advancements promise to reduce latency and delay in such a way to enable high demanding applications. The so-called Tactile Internet [15] supported by the 5G mobile network is working on requirements such as ultraresponsive connectivity, and ultrareliable connectivity in order to enable applications that have strong connectivity/communication requirements. Some of these applications are industry automation, autonomous driving, healthcare, and others. These applications show requirements very similar to those of the DT. Actually, some of these applications could be well implemented by means of DT. From a practical perspective, the linkage between the PO and the LO is effective if the refresh time of the status of LO is lower than the average access time of applications using the LO. This topic is discussed in Sections III-D and V-A.

Another important aspect of the DT is that a wealth of information, that is, the history of the object's behavior, could be stored in order to study it and use it to improve the design, production, and operation of the product in subsequent releases and/or new implementations. For instance, the historical data can help to improve the order management process, as discussed in [16].

The so-called "sequential perspective" in building artifacts, objects, or systems (see [6] and [17]) could be disrupted from the DT concept [13]. In manufacturing, construction phases are well determined and constrained. After a product has been manufactured and initialized, it will go through a highly optimized number of revisions or adjustments for cost considerations. In the case of a DT and a consistent linkage between the PO and the LO, the operation of the physical product could be improved and new releases could be deployed in a timely fashion to better fulfill usage requirements. The DT thus introduces a more agile perspective to the improvement of product functionalities in manufacturing [18]. It should also be noted that many products are now richly equipped with processing, communications, and storage capabilities that could be leveraged in order to support a continuous or more agile improvement of product features.

As stated above, one of the advantages of a DT is its ability to test, experiment with and consider a product under different conditions and usage environments. This reduces the need to always build expensive prototypes and mockups, thereby offering significant cost savings during product development and testing [19]. A side effect of this streamlined approach is the associated reduction in the waste of physical resources. Simulating the workings of a full object compared to actual testing it in a full-sized test facility will bring notable savings in costs, energy consumption, and in the use of materials.

Another relevant feature of the DT concept is the possibility, especially for large systems, such as smart cities, aircraft, large buildings, to functionally distribute virtual systems over different processing environments. These can work as a single large system, and each subpart, subsystem, or component can be fully simulated in several different computing environments. Each subpart could have enough processing power to detect errors, acquire new information, and even to determine how to improve each single part of the entire system by predicting its behavior under stressful conditions. This is a current trend that exploits the edge, fog, and cloud computing capabilities to support the processing requirements of the DT [20].

Simulation capability is, indeed, another major property of a DT system [20]. The behavior, reactions, and issues of a product should be exhaustively simulated. In this way, the high-risk cases of a PO malfunctioning within specific contexts or under particularly stressful situations can be fully covered. In addition, new functions, features, and characteristics of the system could be simulated at will before going into production, and thereby evaluated in order to determine customer acceptability. Grieves proposes, for the DT, a test similar to the one proposed by Alan Turing for the Artificial Intelligent systems [22]: the indistinguishability of the two objects. If a user cannot distinguish a virtual product from a physical one, under appropriate simulation conditions, then a virtual representation of the PO could be considered a DT [6].

Another emerging aspect of the DT is its software programmability. The virtualization of a PO aims at creating a software counterpart of it. In order to fully exploit the DT it is important to have DT's application programming interfaces (APIs) [23] and models for programming the objects [24]. Initially, industries relay on proprietary interfaces and methods specific for their products, approaches, and needs. These specific developments tend to segment the applicability of the solutions. This creates software "silos" of functionalities that limit the general applicability of APIs and solutions. In the case of DT in manufacturing, some attempts to "break the silos" and promote

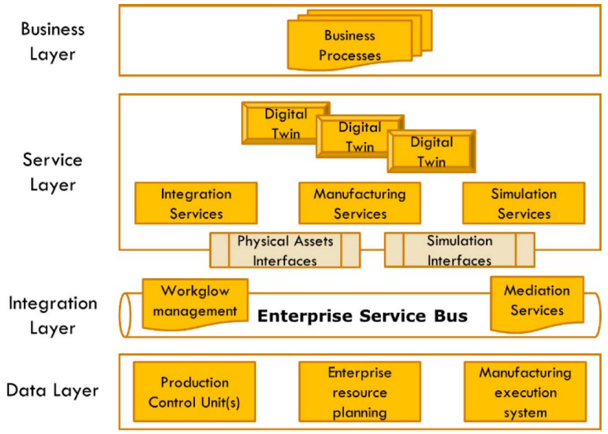

Fig. 3. *Typical architecture of an IIoT platform supporting the DT.*

interoperability are emerging. For example, the IIoT Consortium is addressing the manufacturing community with its architecture and APIs [25], [26].

A great deal of effort has been put into manufacturing for the definition and implementation of the DT concept. This effort covers different stages in the production and also different applications cases [27]. Fig. 3 represents the set of functionalities and their layering for supporting the DT in an industrial environment.

The layering of functionalities is instrumental to the identification and provision of basic services providing different levels of programmability. The data layer represents the different sources and the related enterprise systems that are to be integrated/used during the manufacturing cycle. The integration layer supports the efficient integration and dispatching of well-formed information to all the systems' components. The service layer provides a chaining of services that makes it possible to control how components and services can be created, controlled, and managed. It also provides the ability to manage the DTs and to simulate their future behavior. The business layer deals with the business processes and the business logic related to the production of goods.

From a software platform perspective, the novel manufacturing systems oriented toward the support of the DT concept are characterized by the need to integrate: 1) different flows of data originated by different manufacturing, control, and management systems and 2) several enterprises processes and decision chains [28]. This means that they must exploit Big Data analysis techniques and introduce current best practices for building software solutions, such as cloud computing and microservices [29]–[31]. The DT solution must be integrated with existing systems for enterprise resource planning (ERP), manufacturing execution systems (MESs), and production unit control and their related information flows by means of an enterprise service bus [32]. In addition, the simulation capabilities related to the DT concept are very useful for planning and projecting products [33], [18]. Virtual reality can also be used to visualize the DTs in order to better understand its features [34] or for more complex tasks [35].

IIoT and Industry 4.0 are beneficiaries of many of the possibilities offered by the DT and they are also well intertwined with IoT, and CPS technologies [36], [37]. For this, the DT concept has attracted a wide interest as a possible concept generally applicable in IoT and CPS (see [38]–[44]). For its interesting capabilities, it can be adopted for monitoring the entire life cycle of a distributed system and its objects [45], [46]. It can also be implemented for its capability to represent and deal with a continuous flow of data [47], [48]. This ability is an enabler for data fusion [49]. Dealing with a constant flux of data makes it also possible to apply artificial intelligence (AI) and machine learning (ML) techniques to IoT systems based on DT [28], [50]–[52].

Considering the features of the DT emerged in the manufacturing realm, it is important to look at three relevant issues that may arise in the application of the DT in large open systems.

1) Knowledge of the physical world: It is a daunting task to determine and to describe the models, laws and effects of the real world in an LO. A deeper comprehension of the physical environments in which the PO will operate can be difficult to realize and to represent in a virtual environment.
2) Large systems and products may need to represent a considerable number of parts and their dynamic behavior. Their descriptions and status changes could introduce a higher level of complexity.
3) Siloing/programming: If proprietary and closed interfaces are used, it will be extremely difficult to create ecosystems of DTs capable of working together and being interoperable.

So far, the focus of this article has been on the foundation of the DT as it has emerged in the manufacturing industry and its influence and relation on IoT and CPS. However, the concept of the DT, as it is currently evolving, also inherits some features from other research and technological efforts. These are presented in Sections II-B–II-D with a recap in Section II-E.

B. Contributions of Augmented and Virtual Reality

Virtual and augmented reality techniques [53] have been developed to simulate, describe, and build virtual environments. They also allow to extend and augment the real ones to enable people to interact and act within these different environments by using tools and devices [34], [54]. These techniques also assist people, including those with physical impediments, to interact with virtualized representations of POs in order to play, learn, and to physically act on simulated/extended objects and contexts. Augmented reality is usually associated with the possibility of "augmenting" the amount of information associated with a PO with additional data and to represent the PO and the "augmentation" information as a single entity. The PO is then improved and made more appealing to a user. Virtual reality, on the other hand, tries to create a complete virtual environment [55] in which a user can act and interact with LOs. Sometimes, these objects have a physical counterpart, for example, another user in a game, and sometimes they do not (but they still reflect the expected behavior of POs). The virtualization of reality has pushed toward the definition of the Tactile Internet [56] in which human senses can be stimulated by virtualized objects.

There is special correlation between the concepts of DT and of Avatar within the virtual reality domain. An Avatar is a virtual representation in a virtualized world of an object, usually a person that can behave as a substitute of the physical individual [57]. The Avatar behaves like the intended person and it acts in a virtualized world carrying out actions, collecting data and interacting on behalf of the PO [58]. Games and websites have used these possibilities extensively as part of creating virtual worlds and situations within which an object can have its own life, for example, second life [59]. These applications have been used for entertainment, as well as for training and education [60]. Virtual and augmented reality technologies can be a means to provide improved and more appealing user interfaces for interacting with virtual object representations. Several devices and even smartphones offer the possibility to extend the perception of objects by a human. A visor or multimedia glasses can provide additional information and data that allow the user to better understand and interact with an environment and its objects. Software systems supporting virtual environments [61]–[64] must address several technical issues, including the creation of virtual environments that are realistic enough, the modeling of objects that are part of the environment, their realistic motion representation, video rendering, authoring for objects and applications, the localization of objects and subparts, and many more, for example, such as those depicted in Fig. 4. Several tools and options are available, but the goal of providing an excellent user experience remains difficult to achieve.

From an architectural model perspective, these systems introduce a knowledge layer. It is used for representing objects and events in a realistic environment. Semantics and ontologies are used in order to represent the environment and to reason about it. In addition, a context-related knowledge and the representation of physical laws are needed to govern the interaction between objects. On top of the basic representation of the environments, different tools are used for rendering it at the multimedia level. Eventually, there is an upper layer devoted to the interaction with users and for representing a convincing and intriguing story.

The experience gained in gaming and in the construction of related platforms, even if they are still proprietary and essentially closed, is leading the way toward more realistic

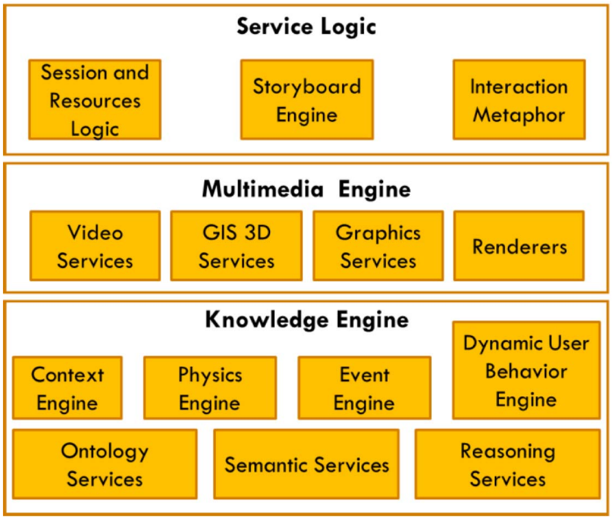

Fig. 4. ***Generic functional architectural framework for virtual/augmented reality.***

and usable applications [65]–[67] that open up the way to several advances in general virtual and augmented reality applications. Some relevant features of the software architecture for virtual and augmented reality distributed systems [68] are especially important for the DT concept.

1) A continuous flow of data representing the changing characteristics of the objects.
2) A continuous tracking of the PO and LO in order to locate or relate them in space and time.
3) Models of behavior representation for POs and reasoning about events and modifications to the context.
4) The representation and the alignment to physical constraints of the physical world.

These features are indeed very relevant for the DT evolution. In fact, POs can change over time in terms of status, behavior, and response to events. These transformations imply the generation of a continuous flow of data that represent the modifications and the current status of the PO. These data must promptly be mirrored by the LO. Changes in the real world occur according to physical laws and the LOs must evolve to reflect them. In addition, POs need to be exactly located in space and tracked in order to accurately embody them. A time-related representation is also needed in order to understand the past behavior of an object and to use the data to predict future modifications and actions.

Software architectures for virtual and augmented systems invest great effort to dynamically reconstruct and display the physical characteristics of the virtualized object so as to provide users with a "natural" representation according to the constraints of the real world [69]. These systems tend to be computationally complex because of two main factors: the processing needed to capture and represent the status changes and the effect of the real environment on the object, and the processing needed to provide a realistic and effective graphical representation of an object. These systems must often process data in real time in order to provide information to users quite rapidly. Gaming is one major example, but education and medical systems also benefit from these technologies. In fact, augmented and virtual reality systems are required to elaborate data so that the timing delay is minimized in order to successfully represent objects and their environment. They can be used in education, e-health [71], manufacturing, and other application domains [70].

The contribution of augmented and virtual reality to the definition of properties of the DT is important. It refers to the possibility to model and represent an LO within a virtualized space. It also creates the possibility to relate the LO to physical characteristics of physical entities in real life. In addition, graphical representation of DT is an important aspect, for many applications, of the usage of the DT.

C. Influences of Multiagent Systems

A technological field that contains many similarities to the DT is multiagent software architectures [72]. These are systems based on the implementation of agents that act on behalf of another entity and explore and collect data in several environments [73]. These agents are used in order to operate, or simulate, or better represent complex environments. Typically, they require the coordination and cooperation of various entities to achieve common goals or tasks.

Multiagent systems offer several interesting properties that a cDT concept may embody.

1) A software agent represents or acts as an external entity that wants to operate in a specific environment. In fact, the agent operates on behalf or in favor of a specific actor.
2) Agents can be of different types, for example, passive, active, or cognitive, depending on the level of intelligence that they represent or collect. Passive agents typically represent objects with minimal meaning for the context to be represented, for example, a rock, a static object, and the like. Active agents are those that are proactive in fulfilling their objectives, while cognitive agents are those that have the ability to apply computationally complex operations, for example, game theory applications or cognitive, ML ones.
3) Agents typically operate in complex systems that are difficult to model. Agents try to apply strategies of different complexity in order to achieve their goals in a constrained software environment. They can put in practice different local actions in order to interact with other agents or with the environment to accommodate their needs. Many agents, especially cognitive ones, are autonomous and operate in a decentralized manner.
4) Agents can collect a large amount of data to improve their knowledge or for backend computation of the environment in which they operate.

There are several multiagent platform solutions, and they have been applied to different applications domains, from security and prevention of natural disaster [74], to e-health [75], smart homes [76], IoT [77], [78], and the DT itself [79], [80]. Multiagent systems are also relevant for their ability to simulate the behavior of an environment, as agents can apply different strategies in order to optimize solutions according to the constraints posed by specific environments.

Multiagent solutions promote a broad cooperation between agents [78]. This cooperation occurs by means of the exchange of information and data between agents, and it is supported by specific protocols or interaction means, for example, APIs. Agents can be numerous and can offer specialized functionalities. Some are focused on a specific application domain task, while others can be specialized in order to offer platform-wide functions to others. Some agents can use cognitive techniques to implement an intelligent behavior. Many AI and ML techniques can be applied in order to achieve agents' goal(s) [81].

Agents must be identified univocally; this is usually accomplished by providing an identity for each agent and the means to identify, address, and refer to it [82]. In addition, agents may be mobile [82], that is, they can move in the application environment in order to fulfill their goals. For large systems, a directory for identifying and managing the different identities and to locate the agents is thus required [83]. In addition, brokering functions are provided to ease the creation of applications that benefit from the optimized allocation and instantiation of agents [84]. In order to tradeoff the usage of agents from the applications and the system perspective, some policing functions can be introduced to optimize how system agents and system resources are allocated and used [85], [86]. These architectures use technologies and methods that are typical of highly distributed systems. A part of the research efforts is tackling new and interesting topics such as how to determine an "agreement" between agents [87], how to assess the trustworthiness of these systems [88], and how to create transactional capabilities [89]. Multiagent systems can also support a relevant function that is at the very basis of the DT concept: the ability to simulate the behavior of a specific environment [90]. The intended goals of multiagent systems are very much aligned with those of the DT concept. Actually, there is a suitable relationship between an agent and an LO. Identity management, allocation, and replication of instances, coordination between different agents, are all issues of great relevance for large DT systems.

The definition of a multiagent systems' architecture has gone through an evolution from the initial widely used platforms, for example, JADE [91], to newer ones with improved capabilities [92]–[95]. Some common trends

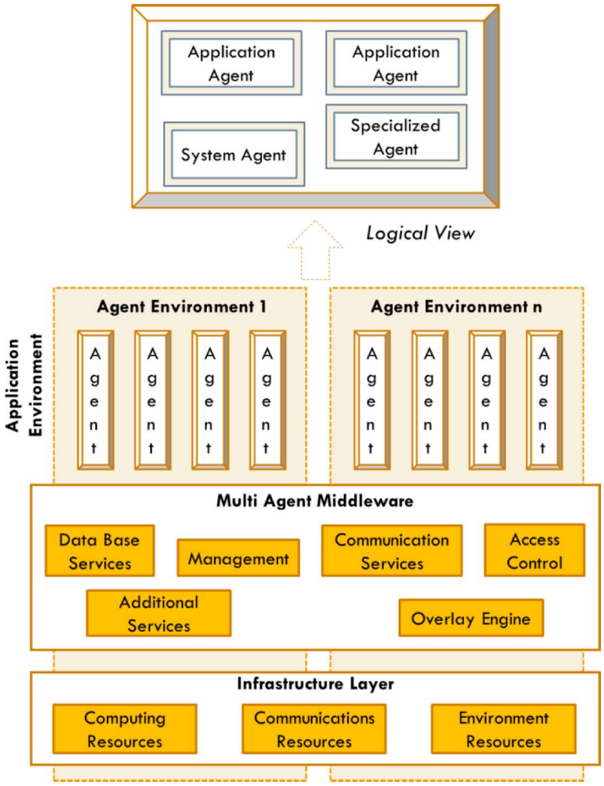

Fig. 5. *Generic multiagent system architecture.*

can be found in these solutions: the layering of functionalities, the definition of platform services, for example, communication and access control capabilities, and the abstraction or overlaying of functionalities.

Over time, the importance of performance has been addressed and the platforms have been tested and improved in order to provide better capabilities [96]. In addition, new concepts have been added to the original multiagent systems, for example, autonomy and self-adaptability [97] and the possibility to integrate multiagent systems with edge computing [98]. These can be seen as additional services/capabilities that a viable multiagent system may provide to programmers and users. The possibilities offered by multiagent systems with respect to simulation [99] as well as to a wide range of different application domains also emerged as early as 2009 [100].

An archetypical multiagent platform is sketched in Fig. 5. There are different abstractions levels, each focusing on specific capabilities and services. The infrastructure layer provides support for processing, communications, or other resources, for example, sensing and actuation. An intermediate level provides general platform services in order to deal with data, control the access to resources, and to support abstraction or overlay capabilities and the like. On top of these layers, specific execution environments, that is, an application layer, are created in order to execute different agent-based applications. On the top, a simplified view of the application is provided: different types of agents with different specific goals are created and cooperate in order to fulfill an application goal. These specific environments can also be used in order to simulate the behavior of large systems.

This type of platform and a large part of issues tackled in the development of multiagent systems are to be considered as valuable contributions for the development of large DT systems.

D. Virtualization Trend

Another technological trend that has a relationship with the DT concept is virtualization. It is the ability to virtualize entire systems by means of software and to execute them on general-purpose machines. It has a relevant impact on the evolution of the Internet [101]. Widely applied in the IT and networking environments [102], virtualization is based on the concept of better exploiting the available hardware by offering the possibility to host different execution environments and related applications on the same set of machines. Usually this trend is coupled with software defined networking (SDN) [103]. Its goal is to decouple the networking hardware infrastructure from its controlling software. Orchestration is becoming an important function in many software systems. It is essential for coordinating and governing the allocation of virtualized resources and for creating slices of functionalities aiming at fully satisfying the needs of specific applications [104]. The recent advances in virtualization techniques have made it possible to virtualize entire systems as well as smaller footprint object containers, that is, lightweight virtualization [105]. Especially in the communications and IoT sectors, this trend has been considered for virtualizing communication resources. There is a current important trend in virtualization at the edge of the network [106]: resources can now be virtualized and provide services and functionalities in dynamic clouds at the edge of the public network. Virtualization has also been extensively applied to the IoT: a sensor can be virtualized in a cloud system and behave as the real one by running software functionalities that represent a specific "smart object" [107]. Objects with limited processing and storage power can also be virtualized and represented in the "cloud," and subsequently replicated several times, freeing the original object to support the processing required to serve all the possible applications.

Another relevant aspect is the possibility to use the layering of software within cloud and virtualized systems [108]. This allows us to clearly focus on the possibility of better organizing the software infrastructure into layers of functionalities that can be reused and exploited by means of APIs.

The reference architecture for the virtualization of network functionalities is the one proposed by the European Telecommunications Standard Institute (ETSI) [109]. Fig. 6 represents the ETSI architecture. The components that provide virtualized functions or are in charge for their orchestration/management are shaded in orange.

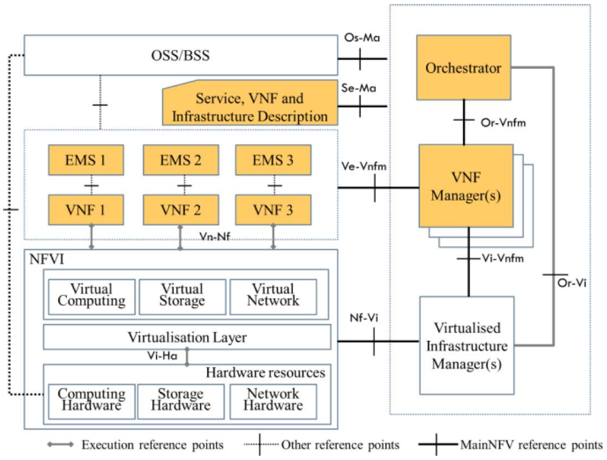

Fig. 6. ***ETSI architectural framework for NFV [109].***

The lower left block represents the network function virtualization infrastructure (NFVI), that is, the set of resources, mechanisms, and functions that allows the virtualization of processing, storage, and network resources. The virtual network functions (VNFs) are on top of the NFVI. A VNF is the basic block in the NFV architecture. It is the virtualized network element that can execute a function or be programmed by means of APIs. The operations support system/business support system (OSS/BSS) block deals with network management, fault management, configuration management, and service management. The BSS deals with customer management, product management, order management, and the like. On the left side, there are the managers and functionalities needed to fully orchestrate the virtualization platform. The NFV orchestrator generates, maintains, and tears down the network services of VNF's themselves. If there are multiple VNFs, the orchestrator will enable the creation of end-to-end service over multiple VNFs. The NFV orchestrator is also responsible for the global resource management of NFVI resources, including managing the NFVI resources, that is, computing, storage, and networking resources among multiple virtualized infrastructure managers (VIMs) in the network. The orchestrator performs its functions by interacting with other managers [e.g., VNF manager (VNFM) and VIM] and does not directly interact with the VNFs. The VNFM manages a VNF or multiple VNFs, that is, it controls the life cycle management of VNF instances. They are the needed management function, such as fault, configuration, accounting, performance, security (FCAPS) function, for the virtual part of the VNF. The VIM represents the management system for the NFVI. It is responsible for controlling and managing the NFVI's computing, network, and storage resources within one operator's infrastructure domain. It is also responsible for collecting performance measurements and recording events.

The layering and the separation of the functionalities of this architecture is designed to better serve programmers and developers in order to add elasticity, higher levels of flexibility, and quality of service in the future communication networks [110]–[112]. It is also possible to further exploit its capabilities to integrate SDN capabilities like service function chaining, that is, the concatenation of different logical functionalities in order to provide a communication service, as depicted in [113]. Network virtualization offer to DT infrastructure the ability to deal with extensive virtualization of functions, for example, an LO, its orchestration and the chaining into different services. These are enabling capabilities for an effective DT platform.

E. Intertwining of the Concepts

These four trends, that is, the original DT concept in manufacturing plus the virtual augmented reality, multiagent systems, and virtualization, provide useful technologies and tools. They have, in different ways, fostered the idea of creating software entities that can fully represent, mimic, and somehow extend the behavior and the functionalities of real-world objects. They point to a large "atoms to bits" transformation [114]. Intuitively, the intertwining of these concepts can create a consolidated and shared definition of a DT. In Section III, the terms, definition, and properties encountered in the state of the art will be melted in order to define a shared view on the DT.

The DT is able to support the transformation of POs (atoms) into software entities (bits). This yields to the possibility to reconstruct a virtual representation of an environment, for example, a vehicle, a factory, or even larger contexts such as cities. This is also strongly related to IoT and CPS because they offer the possibility to measure and determine the status of POs. The measures and the status update can characterize and represent objects and aggregate of them into a software virtualized space. The DT is also usable to influence and retransform the bits into atoms. For instance, in manufacturing, during the design phase, some features of the PO are evaluated by printing 3-D mock-ups of the DT [115]. A sort of "upcycle" is created from physical to logical and then back to physical. This possibility enables a powerful, but not yet well-established concept of virtual continuum [116]. It is the possibility to create a virtual representation of complex systems and to be able to move from the physical to the logical level. This creates a "continuum" between the physical and the logical environments and vice versa. The virtual continuum offers the possibility to control and program LOs creating an entanglement with POs. This approach could have meaningful impacts in many processes and could pave the way to large applications.

The DT concept goes hand in hand with the ability to collect data and information by crawling and crowdsourcing social media [117]. This is an important property of the

Table 1 List of Articles About DT and Some Areas of Interest

Major Topics	SubTopic	List of Papers related to DT
Definitions of DT	Definitions	[1][2][5][6][50][120][122][129][179]
	Internet of Things	[4][40][41][42][43][44][192]
	Life Cycle of a DT	[9][10][18][19][45][46][125]
Verticals	Manufacturing and Industry	[17][27][35][37][115][118][131][185][187][189][193][199][200] [201][268]
	E-Health	[71][75][207][208][209][252]
	Smart Cities and Buildings	[55][138][196][197][253]
	Learning	[191] [219]
	Agriculture	[212][213][214]
	Others	[198][203][204][205][210][215][216][217] [218]
Business Aspects		[13][39][119][120][179][180][184][190][194][195][280][290]
Software Aspects	IoT	[31][36][116][156][175][206][257][263][264][272][275][276]
	MultiMedia	[34][54][68][117]
	Middleware	[20][23][24][25][26][31][142][176][287]
	Simulation	[11][12][33][50]
	Big Data and Data analysis	[28][47][48][49][50][51][52][131][178] [202][211]
	MultiAgent Systems	[79][80]
	Interoperability	[226][227][228][229][230][231][232][233][234][236][237][238]
	IT systems	[266][279]
Industry efforts		[255][265][267][269][270][271][273][274][289]

DT, its description can be complemented/extended by data observed by people and made available into social media platforms.

Table 1 lists these articles that directly contribute to the discussion about relevant aspects of DT within a particular or more technical topics or areas. The table also shows the strong relationship between the DT concept and the general topic of IoT. Definitions, implementations, and use cases of IoT are creating a substrate for a general implementation of the DT concept beyond the so-called IIoT [118]. Verticals are representing some problem domains that have received particular attention from research or that are linked to some of the technical topics presented. The "others" represents a set of interesting applications of the DT technologies in disparate fields.

In particular, Escorsa [119] gives an overview of the current patent effort related to DT. This is particularly important because it shows the attempt within the industry to take advantage of the possibilities offered by the DT implementations and related technologies.

So far, some definitions of the DT are very much associated with specific application domains. For instance, articles [1] and [2] stress out the concept of "product" in the definition of the DT. This focus is very important for manufacturing, but in other problem domains this reduces the area of applicability. Also the concept of life cycle is well posed and described, but it reflects the specific purposes and needs of a particular field of application. Other domains, for example, e-health or cultural heritage, not necessarily fit in the product and its life cycle perspective. Other contributions, for instance [54], focus on aspects of the DT related to the multimedia sector. In this approach, some properties, such as the unique identifier, are generally valid. Others are not necessarily characterizing the DT in other sectors, for example, the 3-D representation of the object. In some cases, the properties listed and advocated as important, for example, sensor and actuators, AI, communications, trust, security, are general ones. They could hold valid for a large part of common applications, but they need some consolidation.

The vast literature mentioned in this section is conceptually aligned on an idea of the DT, but there is not a shared set of properties that can help to create common background, language, and a unifying framework for representing and discussing the DT. In Section III, this article will introduce a set of well-defined properties that can serve the scope to create a consolidated and shared definition of what a DT is. This will be independent of a particular application domain.

Section III will lay the basis for identifying the basic features that are needed to define a cDT.

III. CHARACTERIZING PROPERTIES OF DTs IN THE IoT CONTEXT AND THE REQUIRED TECHNOLOGIES

Numerous descriptions of DT are available [120]–[125] even if they slightly diverge in scope and depth. This section presents a complete definition of the DT concept together with a detailed description of its foundational properties. There is the need to identify a basic set of foundational properties of the DT that can hold for different contexts and situation and still maintain generality. This effort is instrumental to identify a common set of features and naming conventions that can be used to fully describe and specify the characteristics of the DT. This article is an attempt to identify and to normalize a set of properties that can conveniently describe and specify a DT in several application domains. This section offers a unifying framework for clarifying the foundational concepts and providing a possible consolidated definition of the DT.

The available definitions converge at a high level in representing the DT as made out of two entities, a physical one and a logical one. At this level, the DT is an abstract concept whose usage and implementation in IoT software may be ambiguous or too vague. These definitions are not operable ones, that is, they do not define features that a DT must have. In addition, each of the "paths" discussed in Section II looks at the detailed properties of the DT from its specific perspective. The definitions end up to focus on specific aspects of the problem domains, for example, a DT is a representation of a manufactured product, or the DT is a 3-D representation within the multimedia realm without loss of generality. These definitions are useful and they point to right properties, but it would be important to better describe these properties in order to provide a foundational definition for the DT concept. An example is related to a very important property of the DT, its identity. Many articles reference it as a major requirement, a few try to define it [50]. A major statement is that a DT must have a univocal identity. The definitions do not focus enough on the relationship between the PO and the LO. They assume primarily a 1-to-1 cardinality/relation between the assets and the LO. However, this cardinality may also be 1-to-N in case different replicas exist with respect to the asset. There is the need to deal with the DT identity in a more comprehensive way.

Identity of the DT: The PO must be univocally identifiable, for example, by a product code or other mechanisms. The related LO, too, needs to have a unique identifier in order to make it addressable into a software space. If more than a replica refers to the PO, each of them must have a unique identifier and a pointer to the PO identifier. As seen, the LO can be used to represent the PO in time and space. For instance, an LO could represent the engine of an airplane at a specific time in the past, while another replica could represent that PO in the future. Time could also be used to determine exactly what instance of the PO is actually represented by the LO. Similar considerations could hold for the "space." An LO could represent the PO during motion in a specific environment at a specific period of time. This information is relevant in order to fully identify this replica in the lifetime of the PO. Under this perspective, the DT identity can be seen as a more complex aggregation of information, in fact it comprises the unique identifier of the PO, the unique identifiers of each LO related to it and information about the actual time and location of each of the "replicas" in order to represent the specific object in the right context.

Other properties of the DT are differently and, sometimes, blandly defined as well. As an example, many definitions state that the physical and logical counterparts communicate, but it is not clear under which terms.

First of all, some clarifications on the naming conventions used herein: the term DT refers to a PO and its strongly related logical counterparts. For object, we adopt the definition provided by ITU in [126]: "An intrinsic representation of an entity that is described at an appropriate level of abstraction in terms of its attributes and functions." A model of an object specifies functions and services in terms of behavior of the object and of its interfaces. According to [126], objects can represent devices, products, contents, and resources. In perspective, any PO can be represented and virtualized. A DT thus refers to the physical component, the logical component(s) and the relation between the physical and logical entities. The physical entity of the DT can be referred to by a few synonyms such as object, artifact, or product; these terms refer exclusively to the physical aspects of a DT. The logical part refers to the virtualization of the features of the PO and it is usually implemented by software. The logical entity is usually termed LO, digital object, clone, counterpart, reciprocal form, companion, or mate. In this article, to avoid confusion, the term DT will indicate both physical and logical components, as well as their relationship. When referring to physical entities exclusively, the terms used in the literature are many, for example, artifact, product, or more generic terms like entity or object, and, in those cases. The adjectives physical or real precede the substantive term. For logical entities, the terms used are clone, companion, or duplicate, or even other terms that identify objects that are derived from physical entities. In these cases, the adjectives logical, virtual, or digital, or the name of the substantive software can precede the term in order to avoid ambiguity. In this article, PO and LO refer to the physical and the logical parts of the DT. The term cDT is used to qualify this attempt to consolidate the different facets of the DT concept as a result of the fusion and better specification of the different requirements, technologies, and numerous interpretations/ definitions.

The cDT can be defined as the constant entanglement between an artifact (the PO) and its software representations (the LOs). The DT links two different entities, a real one that is relevant in the physical world and a softwarized one that is executed in a virtualization space. The nature of the real object tends to be physical, that is, a building, a sensor, and a human. However, some real entities could be software or immaterial as well. Imagine, for instance, a concept like Boolean logic; in principle, it could be represented by a digital counterpart that describes the concepts of that theory. In a recursive way, an LO could be the "real object" associated with an LO and so forth. For the sake of simplicity, in this article the originating object is material in nature.

In order to implement the concept of cDT, the LO(s) [i.e., softwarized object(s)] needs to be supported by a software environment, that is, a computational and communications environment tailored to the specific needs and objectives of the DT representation. This environment must comprise processing and storage capabilities as well as communications to support the mirroring of the PO and the constant exchange of information between the PO and the LO. Furthermore, this virtualization space can offer additional capabilities and functions in order to protect and ensure the life cycle of the LO. This environment could provide allocation and orchestration functions of all the resources required for guaranteeing the internal DT interactions and coordination, as well the possibility to be integrated with other objects and systems. In order to better describe the cDT concept, a set of essential properties of the DT must be clarified. An initial set of the qualifying properties of a cDT, derived from the state of the art, is presented in Sections III-A–III-L, and summarized in Section III-M.

A. Representativeness and Contextualization

Generally, the LO has to be as much as possible verisimilar to the original; however, representing a PO in all its facets and implications is difficult and sometimes worthless. The cDT should be supported by a model designed and implemented with a set of goals and purposes, and refer to a target context in which to operate. The LO should at least represent those properties, characteristics, and behaviors that are necessary and sufficient to qualify the LO as representative of the physical one under all the intended perspectives and features to be analyzed. A PO is described by its attributes [127], properties, and behaviors. One of the basic concepts of the DT is related to how much the replica represents the original object. The definition of a model representing the relevant characteristics and behavior of the PO is the objective of any cDT. Representativeness should be considered under three major parameters.

1) Similarity, that is, how much and how well the LO reproduces the original object and its status and features.
2) Randomness, that is, the probability that the LO, the replica, has a different status or is providing diverging features from the original one.
3) Contextualization, that is, the two previous features must be considered in the context of the operation of the correlated objects. If the usage context of the DT is a specific environment, most likely only a subset of all the features, properties, and information of the PO are relevant.

In many application scenarios, some of the attributes of the PO are not relevant, that is, they do not characterize or influence the behavior over time, or the states of the object for the intended purpose of the DT. In this case, they are not considered in the description of the LO. Contextualization means that all the relevant features and data available are needed and sufficient to represent the PO in the specific virtual space under consideration. Modeling of the DT is still a difficult task; however, practical methodologies are emerging [17].

B. Reflection

All the meaningful attributes, features, status, data, events, actions, and all the other information characterizing the PO are to be timely represented by the softwarized LO and vice versa. For the time being, considering the IoT application domain, this property refers to measurable aspects of the PO. These measures are then represented by a set of values that can clearly reflect the specific status of the PO in the analyzed context. The LO is punctually embodying these measures and then it reflects the PO status. A complex physical system, for example, human body, is not easily representable as a set of variable and attributes. They sometimes can only be partially "quantified." Modeling of the PO is one way to determine, or to decide, which aspects to focus on. IoT technologies can help in quantifying a part of aspects of the PO. It is important to timely understand if the model is representative enough of the PO. Reflection property refers to a PO that can be accurately measured and represented with respect to the application goals. Under this perspective, a PO is described by an LO as a set of values, related to status, attributes, and behavior. This information may change over time. The PO is fully described if all these values are timely mapped on a mirrored set of the same values describing the LO. The reflection capability of the DT suggests that each relevant value of the PO is univocally represented in the mirrored object. There may be several transformation functions that relate the values of the PO to the values of the digital reflection(s), that is, the LOs. Let us assume that a PO is fully described and characterized by a set of variables and their values

$$X = \{x_1, x_2, \ldots, x_n\}$$

in a specific multidimensional space S where $\forall x_i \in S$. The reflection properties state that

$$\exists f(X) = X'$$

where

$$X' = \{x'_1, x'_2, \ldots, x'_n\}, X' \in S, \text{and } \forall i \, x_i \equiv x'_1$$

(the sign $\equiv$ indicates that it is equivalent or congruent). In this sense, the LO fully reflects the salient features and characteristics of the original object. Hence, the DT representation holds.

Actually, the function $f(X)$ could be generically an equality function, or in some cases, it could transform the values of X into a different X' that is congruent to the original set. In other cases, more than just a transformation function could be needed. In these cases, the combined function set from X to X' is injective, that is

$$\forall i \in S, \exists! x'_i \text{ such that } f_i(x_i) \equiv x'_i.$$

Fig. 7 shows a case where different transformations can be applied to the LO. In this case, the relationship between X and X' is injective. Injection is not prescriptive, in the sense that more complex relationships can be established between the sets. For instance, when applying AI techniques, different values and features may contribute to determine a single value of the LO.

The reflection property also points to the fact that the PO is placed in time and space. A typical structure for representing a PO attribute can be seen as a set of triplets. Each triplet could have the following format:

$$\text{attribute}_i = \langle \text{timestamp}, \text{location}, \text{value} \rangle.$$

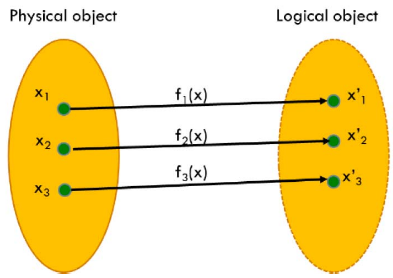

Fig. 7. ***Relationship between the PO and the LO by means of functional transformations.***

C. Replication

This is the general ability to replicate an object into a different environment. A PO can be virtualized and replicated several times in a virtualization space. Essentially, POs can be softwarized, that is, cloned, several times, and each LO can itself be replicated as well. Fig. 8 depicts some replication patterns for objects.

The case of a PO replicated three times is represented on the right side of Fig. 8. Each LO reflects the status of the physical one. Over time, the set of characterizing attributes of the PO and the replicated logical ones should be consistent. The left side of Fig. 8 shows the case of the three LOs in synch with the physical one; on the right side, an LO acts as a master replica. It is in synch with the PO and it provides synchronization to the other two LO replicas of the PO. This master replica is responsible for the synchronization of all the relevant information.

Replicability is assured by the ability of software environments to virtualize components. Virtualization offers a wealth of opportunity. For instance, virtualizing a sensor in a software-powerful environment, such as a cloud system [128], allows the "small" physical device to be freed from coping with multiple polling requests. Functions executed in the cloud help in reducing the local processing burden at the PO level. A single virtualized instance (i.e., an LO) can have enough processing power in the cloud to reply to multiple and real-time requests. In addition, different replicas, each devoted to a single application or domain of applications, can be instantiated and fed by a master LO or by the PO itself. Each replica can be tailored to the needs of the requesting application. Replicas can also cooperate to share data and information about the PO quickly and efficiently.

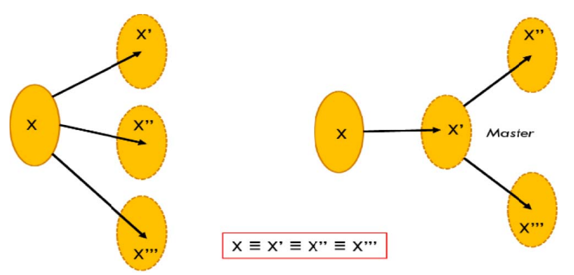

Fig. 8. ***Examples of replication patterns for virtualization of LOs.***

In principle, replication from virtual to physical is also possible, for example, POs may be replicated [122] and associated with a master LO that keeps track of the functioning of the POs. This should not come as a surprise; in fact, smart manufacturing is moving from the design of LOs to the actual implementation of several POs that can be coupled with one or more corresponding LO(s). A practical example of replicating DTs in the context of a CPS is provided by [129]. In addition, an LO may also mediate and represent the change status requested by applications.

D. Entanglement

The DT concept represents the linkage between a PO and its logical one. This means that all the information that fully describes the object must be passed to the logical replica, and in real (or very close to) time. The logical replica makes this information available to the applications and services.

This communication relationship is termed here entanglement because it refers to the instantaneous exchange of information between two closely related entities. This entanglement characterizes the PO and the LO and so at least three properties must be considered.

1) Connectivity: There should be a direct or indirect means to communicate the changes of status and related data between the physical and the softwarized LOs. Depending on the type of (physical) objects, they may or may not be able to process, store, and communicate data. If these abilities are present, the PO can forward, by means of networks and supporting protocols, information that fully represents itself to the LO. Transmission can be direct, that is, the PO and the LO are capable of providing direct communication; or indirect, that is, the two communicating objects relay on a third party for sending and receiving information. Not all of the POs are capable of communication or processing. In this case, communication can occur by means of other objects that are capable of directly observing and determining the status of a PO and then feed that information to the LO of the observed physical entity. For instance, the status of an object can be monitored by means of a camera and/or sensor, with any relevant information extracted from the multimedia flow and passed on to the LO.

2) Promptness: The exchange of information between PO and the LO should be timely, that is, in such a way that the time between the changes of states of the PO is negligible with respect to the needs and intended usage of the LO by applications or users. For instance, if the entanglement is between a physical parking lot and its LO, the information about the parking lot occupancy should be exchanged in less time than the time needed for a car to enter or leave the parking. This should give a meaningful representation of the actual occupancy. For some POs, real-time processing, communication, and storage capabilities could be required in order to properly keep track of the events and changes in status, while for other objects daily updates or even longer periods may be acceptable. As a rule of thumb, the average time elapsed between two changes of status should be intended as the upper limit interval for sending updates to the other object. For some objects, a very short interval for synchronization may be a stringent requirement for the use of the DT approach, for example, medical applications, robotics, industrial Internet applications, and the like.

3) Association: The relationship between PO and LO can be unidirectional, that is, from the PO to the LO (e.g., a sensor sending data), or from the digital to the PO (e.g., the LO of an actuator is sending commands to its PO); or bidirectional, that is, a continuous exchange of status information between the objects. Typically, the intended direction of communication is from physical to logical. However, for POs that are instrumented, there could be a great value in supporting bidirectional communication. The PO could provide relevant status information, and the LO could provide relevant updates and adjustments to the physical one to improve its functioning.

Entanglement is a fundamental property that strongly characterizes the concept of DT: strong entanglement occurs when the PO is constantly linked to the logical one. The link is bidirectional and the LO has the ability to modify or update the status of the PO. In other cases, the relationship may be defined as simple entanglement, that is, the communication is unidirectional or it is not real time, or the linkage may be interrupted for a certain time. Another form of entanglement can be considered: weak entanglement. This form of association between a PO and its LO can be established when data and information about a PO are inferred and derived by the analysis of data stemming from the environment around the specific PO. This may occur by observation, by interpolation/calculation or by crawling the data from social networks or by analyzing other objects' status/values. This information is characterized by the fact that it is not always available, it is generally not fully trustworthy, and it is not necessarily available in a timely manner. However, new AI technologies can help in acquiring and analyzing data from several sources and thus infer and even predict accurate information about a target object [130]. The issue of entanglement and the fast data acquisition is an important one for the DT as well as for IoT systems. For example, in [131], a practical example in the context of cyber physical production system is given.

E. Persistency

This property refers to the fact that the DT should be persistent over time. Actually, the PO can have real-world limitations that restrain its functioning. The LO should be able to compensate and mitigate these limitations and to support a constant availability (and serviceability) of the DT. The LO within the DT is the main enabling factor for this property, and it has to be persistent and resilient in order to be always available. It is the main instance within the DT, and its states and values should be the reference values for the applications. In the case of malfunctioning or other problems with the PO, the LO should be the source of information for re-establishing and synchronizing the PO to an acceptable and meaningful state. Different approaches can be adopted to guarantee the persistency of the digital copy. In fact, different levels of replication and management functions [132]–[135] can be introduced in a DT system, as well as different communication capabilities and related platforms functions in order to implement a resilient DT framework [136]. While all of these mechanisms ensure the high persistency of objects and memory in large distributed systems, they do introduce the issues of managing different copies and their consistency. Issues that may compromise a prompt access to data are discussed in Section V-A, where an example of solution is depicted. Similar issues at the edge level have been considered and analyzed in [137].

F. Memorization

The properties of contextualization and representativeness introduce another important feature of the DT, that is, the ability to store and represent all the present and past data relevant for the DT. These data characterize and describe the past behavior of the DT. POs interact with very complex environments and are immersed in them. They reflect the dynamics of the physical context in which they are operating [138]. Objects are subjected to the laws of physics as well as to social and human laws, habits, behavior, and attitude. Certain objects can be given meaning and value in addition to their tangible one, for example, some objects like a cloverleaf, can be a sign of luck or other human-assigned properties.

The DT concept brings with it an important issue: how much of the complex context in which the PO is immersed should be considered by the digital copy? What are the relevant data to be stored? If the object (e.g., the cloverleaf) is going to be used for a specific goal (feeding some animals), how much should it be contextualized with respect to the envisaged application of the DT? Should the entire context of a PO be recreated and represented? In principle, the DT should keep a set of all the meaningful data together with their location and time indication. In this way, if an object changes, the meaningful features of this change will be stored and the object can be analyzed in a specific period of time while considering its several locations (its "context").[1] However, the relations of the objects with their environs

[1] Actually, it is also important to create a representation of the context [139] in which the object and other entities operate and have operated in order to be able to recreate and study the past situation of the entire system.

(that may be useless and neglected now and thus irrelevant for the current technologies) could become very important and meaningful in the future.

It is important to collect and store (or be able to calculate and infer) as much as possible data. This abundance approach is mandated because DT should capture all the facets and all the features and relationships of their POs with respect to the contexts and environments in which they operate. As has been described, in certain cases, the actual usage of a DT by applications involves the need to access, check, and operate on a limited set of properties of the entire spectrum of the object's features. For instance, if an application is using a DT to monitor a home radiator, most likely its only valuable information is related to its temperature and internal pressure. Properties like color, size, etc., may have little or no relevance for the context of the "Home Application." It is essentially up to the applications to select the aspects and facets of data that are meaningful for the goal at hand among all that are available and stored. However, the DT should be programmed for abundance and completeness of its associated data set. Large data sets spanning different object features are difficult to manage without an emphasis on the importance of specific data [140]. Still, it is important to store and preserve a large amount of raw data that could be better used with future techniques and tools. The DT data sets are to be used in two ways: to understand the behavior of the object within its operational space, and to predict its possible behavior in the same space or in other environments. Contextualization in this case means to organize data in such a way to be able to represent, discover, and manage new dimensions or relationships of the LOs with its environment. Considering that POs can last for several years, there needs to be a way to properly store and manage this amount of data in an open fashion respectful of privacy and of the ownership of people's data [141]. The quantity of historical information, for certain objects, increases over time and there are also new findings and new contextualization of them. This can lead to new discoveries and identification of new relationships between. "Objects with memory" [142] is a technology that tackles the issue to save the history of the "old things" in the context of IoT.

G. Composability

In real life, objects are often an aggregation of different entities. The composability of a DT, that is, the ability of grouping several objects into a composed one and then to observe and control the behavior of the composed object as well as the individual components. If the DT is to be used in large systems, then there must be a way to widely and efficiently support integration and composability. For instance, several POs in a building, each representing a different subsystem, for example, heating, control, water, electrical, and other systems, may be composed into an individual well-formed representation of a large system, for example, a smart building, as a whole in order to determine and control the behavior of the larger entity. In this case, a complex aggregation of POs is occurring in the real world. In the virtualization space, the LOs can represent a subsystem or actually the aggregation of POs that comprises the subsystem. Depending on this choice, the complexity and the granularity of control of the DTs can greatly vary,

In general, objects can be seen as groupings of subparts (subobjects) or as the combination of several individual objects. A car could be considered as a single object, but it can also be seen as the aggregate result of a combination of different objects. Each object, for example, the brake system, the transmission, power production, etc., fully interacts and cooperates so that the car executes its tasks as a whole (as a unique entity). Each single system or subsystem can be seen as an individual PO, and as such it can be represented in a digital form. Therefore, it is important that all the LOs representing parts of a larger entity can be represented, considered, and interacted with as a single LO according to the needs of the applications. Composability also represents the ability to abstract the complexity of a large system and to focus on a few relevant, for specific applications, status, and behaviors of the entire system without having to consider the functioning of all the aggregates' subsystems.

In order to support the composability of objects, several software engineering technologies must be combined and utilized. From the definition of composability in software [143] and some implementations [144] up to the software component model [145]–[147] efforts, software engineering shows a constant trend to create conditions and frameworks for supporting component integration and communication, for example, also in specific fields like CPS [148]. The efforts related to virtualization also offer mechanisms and technologies for supporting the composability property of the DT: microservices [149] and containers [150] are possible candidates for supporting the integration and composition of several LOs. Orchestration is an essential function [151] in order to govern the aggregations. Simulation theory [152] and agent-based simulation technologies [72], [153] for large systems also play a key role in representing the behavior of a large system composed of different DTs. A DT will also represent the processes and activities that need to be fully considered, in this case multiple technologies and representations can be used to characterize, identify, manage, and improve the internal processes [154]. A large composition of DTs could become the definition of a large system of systems, and hence reliable approaches to complexity are needed [155]. Some middleware platforms for DT are developed [31], [156] and are aiming at composability. Different studies and experiments have addressed the need for composition/aggregation of objects within the IoT context, for instance, the EU project iCore has provided and experimented the possibility to virtualize and aggregate different objects [157]. The needs for aggregation and a model for supporting it for the DT in an IoT context are presented in [158].

H. Accountability/Manageability

This property refers to the ability to accurately and fully manage DTs. While POs can fail or break, LOs should not "break," instead, they should enter into a recovery state in which they are still capable of responding to queries about the physical counterpart and to show all the latest important functional values. LOs should also apply policies and measures to limit the impacts and the damages to POs. The LO can be seen as a "flying recorder," that is, it is a trustworthy recording of all the states of the PO. This feature could also be used to understand and recover the latest states of a PO and to resume its operation.

A PO may be subject to management and accountability processes that should be fully replicated by the logical one. The LO should also guarantee its existence beyond the lifetime of the real object and it should be possible to manage and execute it as a virtual entity within a highly distributed software environment. These requirements point to the manageability of the PO as well as the manageability of an LO, a software entity, within a virtualization space. In addition, due to the composability property, the LO should be part of a larger and complex system that can be mirrored and monitored/managed as a whole. Management techniques for large virtualized infrastructure [159] could be usefully adopted in these cases. The concept of slice as elaborated for 5G virtualized networks [160], [161] can be used to create specialized environments that are compartmentalized and devoted to specific tenants, that is, softwarized virtual environments that reflect the operation of large composed physical environments. For instance, a smart city application has been prototyped exploiting these concepts [162].

The required processing, storage, and communication resources that virtualize the physical ones are allocated in software systems, from where they can continuously monitor the physical resources and verify their behavior against the intended policies of the system. In addition, in the case of failure of the LO, while the physical one maintains its operativity, there must be a way to quickly restore and resume operation with minimal loss of state information. Recent techniques for rapid recovery and restoration of virtual machines [163]–[165] can be adopted to provide these required capabilities. For instance, some experiments have been carried out in [166] and [167]. Moreover, the same LO and its components may be made available to different users, and so there is the need to support multitenancy and shared usage [168], [169]. Multi tenancy is in fact demonstrated in the vIoT testbed [170].

Due to the possibility of applying the DT concept to several application domains, there must be a way to guarantee different levels of manageability and accountability. For instance, a digital patient needs the highest possible features and functionalities, while other applications may need a much lower level of services. The notion of self-organizing (self-X) systems is also influencing the definition of the DT [171], [172].

I. Augmentation

POs come with well-defined functionalities and services that are fixed for the entire life cycle of the object. Even if they do not have processing capabilities, they may have limitations due to constraints and costs related to manufacturing processes and materials. However, the DT can leverage the software dematerialization: the LO can, actually, modify, update, and improve its functions over time. In other words, it could be functionally augmented, that is, new functions and features could be implemented in the LO. There is a long history of the computer augmentation of POs [173], [174]. More recently, this trend has been applied in new domains or it has profited of improved technologies, for example, in big data and smart manufacturing [28], and augmented and virtual reality [175]).

Obviously, new features are software-based and take the form of innovative and more intelligent functions enabled by APIs or by the analysis of data sets related to the PO. By means of APIs, a set of POs, through their LO counterparts, can be made interoperable within a complex environment in order to cooperate and achieve specific business results [176]. A PO, for example, a statue, could become programmable, for example, by offering applications the possibility to access data related to the PO (materials, construction, authors, historical information, and many more). In addition, if the PO has processing properties, then the LO, and hence the DT, could be related and could actually interconnect and interact with other objects in order to better achieve their common goals and objectives. Augmentation can thus be achieved by using the DT's data and by the exposure of APIs for controlling, governing, orchestrating, or simple-querying the DT.

In this area, the EU project iCore has provided some practical implementations and a seminal architecture [157]. The possibility of defining platform mechanisms to support the creation of services is also discussed in [177].

J. Ownership

Ownership is another important, but often neglected, property of a DT. It is declined in two different ways. The first one is related to data ownership. The DT, as many IoT systems, produces a large quantity of data. It is important to determine and regulate the ownership and usage rights of these data. For example, Rødseth and Berre [178] proposed a data management model for the data produced by DTs in the maritime industry. The second way refers to the ownership of the DT and, in particular, of the LO. The POs have, typically, an owner. Their replication can create a set of LOs that refer to the physical one, but they not necessarily share the same ownership. For example, the copy of a painting, a photograph, or other multimedia material can refer to the original object, but the ownership could be different.

The industry is particularly interested in changes of ownership and its management for business reasons [179], [180]. In addition, some LOs can offer different capabilities of interaction or better ways to exploit the features of the PO and, consequently, they can have different constraints in ownership. For example, a rent-a-car company can offer its DT applications to users, while a brand maker can directly offer its own and temporarily associate it to the rented car. Ownership may pass from one object to the other. Augmentation can also have an impact. In the case of a painting, in fact, a user can use the APIs of the LO to transform and to mashup the original representation creating a new original digital artifact. In this case, the ownership of the original PO remains the same, but the ownership of the mashup should be of the final user (that should pay royalties for the original object usage). Ownership could also be useful to understand the story beyond a PO, a used car, and to fully represent the status of the artifact. Ownership may be a complex issue, but the proliferation of DTs needs to deal with this property in such a way to introduce flexibility in the logical representation of the original PO.

K. Servitization

The DT concept is essentially based on the ability to create a software clone of a PO. This LO, as already discussed, can be augmented with software interfaces in order to control, govern, or simply get data out of it. The augmentation property also aims at increasing the number of functions that an LO can provide [181]. Thanks to strong entanglement, these functions can be used to control and to operate on the physical side of the DT. These capabilities are instrumental to the ability to create a large number of new services and functionalities on the entire DT. In this case, the DT is a means for offering high level of Servitization [181] of a PO. Servitization refers here to the ability to offer in the market the association of a product with services, functionalities, processes, and access to data of a PO by means of software capabilities, tools, and interfaces. These features complement and characterize a product that has moved from being merely a "good" to become a set of services acting upon the "good." In fact, the product is now seen by the customers in terms of its functionalities and not only as an object. A permanent linkage between the customer and the producer is then established by means of the ownership of the servitized product. Servitization is a term and a set of technical definitions and methods that are very interesting to a broad industry and business community [182]. Obviously, there is a great interest in the manufacturing community for the concrete implementation of strategies and processes related to the servitization of products and their DT. Possibilities range from personalization and customization to the definition of a compelling set of software functionalities that will complement and augment the product and will be the major added value of a physical product [183]. From a business perspective, servitization is a great possibility for transforming existing and future products into services appealing to customers. In the long term, servitization could lead to a new kind of economy no longer based on the ownership of things, but based on a pay per usage philosophy. The DT concept is a major enabler for this kind of change. Industries and especially manufacturing are extremely interested in servitization. In [184], an analysis of some current industrial initiative is provided as well as a general overview of the "servitization of the DT."

L. Predictability

A DT represents large data sets of events and properties. It is intended to operate in known, well understood and embodied contexts. It is capable of interacting with other objects. The predictability property refers to the possibility of embedding an LO of a DT in a specific environment and to simulate its behavior and interactions with other objects in the future or during specific period of time [6], [20]. This is a growing trend of AI that can also be applied to the DT concept [185]. This is an important signal of the need to create large, continuously running complex systems devoted to the prediction of DTs' behavior in particular situations or environments [33], [186]. IoT systems and DTs can even lead to the control and simulation of the behavior of large complex systems like cities, factories, logistics networks, and the like. There are examples of usage and experimentations of the predictability of the DT in the industry. For instance, in [33], DTs are embedded in Virtual Testbeds in order to execute several experiments. Also, NASA is using the DT for its own developments [187]. In [188], a survey on current efforts within the IoT field to use simulation techniques and, in particular, the DT is given.

M. Short Recap of DT Properties

These properties, some of them are represented in Fig. 9, hold for a very rich and comprehensive DT concept and for its supporting system. This system should be capable of fully representing the POs by means of LOs and their entanglement, and reflection properties, to control and manage them. Additional capabilities such as memorization and predictability enable the projection and prediction of the behavior of DTs in the future.

Some of the properties discussed herein are foundational for a DT, that is, without them there is no real DT implementation; while others extend and increase the intrinsic value of the DT relation.

The first group includes the following properties: reflection or mirroring, virtualization, and entanglement, along with representativeness and contextualization in space and time. They ensure that the LO fully represents and behaves like the PO within one or more specific operational contexts. With these properties, a DT system offers the

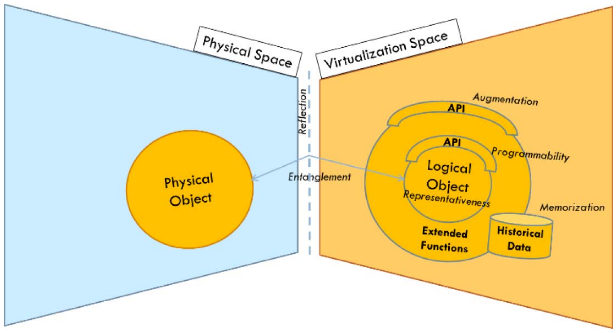

Fig. 9. ***Representation of some DT properties.***

ability to represent the current status of the PO and can be considered as a basic implementation of the concept.

The next level of richness is achieved by introducing other properties. Composability and accountability/manageability introduce the ability to interact between different LOs, softwarized objects, and to compose them into larger aggregations. This offers the possibility of accounting for their usage and behavior as well as introducing self-X management capabilities. Memorization guarantees that a DT can be represented and controlled along a large part or the entire life cycle, while augmentation ensures that it can become a programmable object. Ownership introduces the ability to correctly manage intellectual properties rights issues and associated responsibilities of the LOs. Servitization is a property that ensures the usability and the effectiveness of a DT with respect to its usage by final or intermediate customers. Predictability is an important property that ensures that the behavior of the DT within its operational contexts can be studied and predicted.

An important issue is to ensure the validity of the properties listed in this section. This article has referred to, analyzed and represented relevant literature about the DT definitions and usage. Some properties are generally considered valid, for example, identity, representation in [50], identity, communication, and others in [54], entanglement is substantially addressed by the definition of Tactile Internet [15], and extensively used in the practice of the implementation of the DT. Others are considered useful and they have been mentioned or introduced in some experiments or implementations. The identification of these properties has also the objective to look ahead in the future. If the DT concept will be implemented in several application domains then these properties can result a valid means to reason about the DT and its features. A goal of this article is, in fact, to define a common understanding and description of the features of the DT and to allow a large community to point to the common concepts and features of the DT. As a first step for determining whether these properties are valuable, some scenarios will be illustrated in Section V in order to show their usefulness. Actual implementations will also streamline the definition by showing the usefulness of some of these properties. In fact, not all of the identified properties will be fully implemented in an actual DT system, and a natural selection is expected, as well as the practical identification of other important properties. However, listing them, while the DT usage is on the rise, can be useful to understand to what extent a particular implementation realizes the enactment of a cDT. Actually, a fully fledged DT system does not exist yet, and it is not sure that one will be implemented soon. However, the definition of these properties is important in order to understand the level of functionalities and aspects covered by specific implementations.

IV. VALUE OF THE DTs CONCEPT

The DT concept has been developed in the manufacturing industry with specific goals and its importance is likely to increase in the future [189]–[193]. The concept is also appealing for other application domains, and, along with the extensions, "interpretations," and consolidation discussed in Section II, it has been studied and partially applied by several research efforts and in different industrial contexts. Some industries are interested in the concept and looking for potential applications [194], or checking the costs and applicability of the concept [195], or considering how to cope with data collection and usage [131]. There are ongoing technological trends for creating DTs in several fields: smart cities [138], [196] or specific issues of the cities, like traffic [197], aerospace [187], aircraft fleet management [198], innovative factory [2], [199]–[201], environmental management [202], logistics [203], [204], the future of work organization [205], the IoT and CPS [36], [206], healthcare [207]–[209], asset management [210], predictive maintenance [175], [211], farming and agriculture [212]–[214], power control systems [215], and many more. Even some unconventional ones like [216]–[219] are emerging. The concept has also attracted the interest of governments with possible funding and industrial fallout [120].

The DT has been implemented in different applications in several fields (see Table 1 for a recap of some of the application areas addressed by this survey) and, most likely, it will receive attention from other sectors. It is important to discuss to what extent the DT concept is generally valuable. And, from a research and applicability perspective, under what constraints and approaches can it bring value to solve implementation problems. More specifically, is the DT useful for the IoT problem space? Should it be considered as a basic concept to be used in IoT applications? Or should it be used mainly for specific limited issues and problems?

There are a few typical scenarios for the usage of the DT that can be inferred from the state of the art.

1) Design and consolidation of products, that is, where the DT is used to help in the design and the production phases of complex products and then used

as a means to collect and check the operation of the product in order to identify variations or unexpected behavior.

2) Prediction and simulation of the behavior of an aggregated set of DTs in order to understand, control, govern, and orchestrate the behavior of a complex system. This is supported by the collection of historical data that show the past behavior of the PO.
3) Servitization of a physical product and its augmentation in terms of new functions and interaction with the customers.

Clearly, large IoT systems, such as manufacturing or smart cities, could especially benefit from the unique properties of the DT concept.

A. System Value

It is extremely difficult to determine exactly which types of problems can benefit the most from a DT-based system. The manufactory model that generated the concept is a good example of its applicability and the kinds of complex problems it can help to solve. The replication of a set of objects' behaviors in a well-formed software environment can help to determine its faults and/or possible improvements. Two specific issues need to be considered: the need to properly represent the set of objects and the need to properly characterize the environment in which the LOs will interact and operate. This requires a good set of modeling capabilities as well as the ability to fully understand the constraints and limitation of the POs and the environment in which they operate. The object, or a set of objects, and the environment in which they will operate should be modeled in an autonomous fashion [220]. A full description of the objects and their environment can be a daunting task if completeness and exhaustiveness are required. On the other side, a poor description can be detrimental to the entire system and it can generate inadequate data and results.

The application of the DT implies a sort of world view: the description of the environment and the actual phenomena is realized in terms of representations of the involved objects and their interactions. Some objects are aggregated in order to create a larger entity that reveals its external behavior as a consequence of the behavior of many other components. Grasping all the relevant features of the involved objects and their roles as well as their internal interactions is a significant modeling effort. Object manipulation, and in general, executing or determining actions over a set of cooperating objects must be supported by an interaction model, for example, [221] for robots' manipulation of objects. The complexity of the models increases in relation to the number of involved objects, and so very large systems' modeling could require a huge effort.

In certain cases, it may be very difficult to associate some data, for example, those collected by a sensor, to a specific PO. A temperature value could refer to a specific object, and/or to an entire environment. Actually, a sensor can represent an object, or some properties of an object in its environment, or directly properties of the environment in which the object is embedded.

Another important issue is related to when applying the DT approach. In fact, many problems can be solved without introducing the DT concept. For instance, in large cities, traffic information could be relevant independently of any specific car that may be generating information or is involved in a traffic jam. Many "simple" IoT implementation could use these data to offer very effective services for traffic detection and prevention. However, the DT representation could be used in order to provide more granular information. For instance, which cars are involved in a traffic jam, where each of them is heading, and how much a single vehicle is polluting the environment. Having more detailed information by means of a set of DTs could enable the provision of better models, applications, and solutions to the people involved.

The most tangible value of a large DT implementation may reside in the possibility to observe, analyze, and understand real-world interactions and impacts on different objects at a very granular level. Such a system could also offer the possibility to predict and actually simulate objects' behavior under different conditions. This analysis may be difficult to achieve with other means. Wherever the unpredictability of behavior and complexity in interactions and in the change of states are relevant, DT modeling could be a viable option for attacking some control problems.

The DT approach could be compared to object-oriented programming, a generally accepted programming practice that has been widely used. Over time, extensive studies have been conducted to better understand how well this paradigm can represent the world in a computational way (see, for instance, [222]). Following this analogy, DT may not be the right choice if the problem at hand has a very procedural solution, or if it can be solved in a functional way.

In addition, the implementation of the DT concept suffers from the lack of an established set of platforms, modeling, and development tools that can scale up to very complex representations. Having a large-scale platform would reduce the time required for implementing solutions, so that programmers could focus more on representing the DTs than on developing a solution for supporting that representation.

B. Interoperability Value

The DT approach is sometimes based on a "closed-loop" approach. The focus is on representing a context and its performing entities for a specific purpose. This creates silos of interoperability, that is, specialized systems that offer some level of internal programmability. They are efficient and effective for the immediate goal and tasks, but they pose issues in interoperating with other systems. This approach is, partially, shared with some IoT/CPS developments that are focusing on specific problems and environments. There are several initiatives in IoT realm to favor the interoperability of different systems and a strong push toward interoperability, often by means of standardization. For instance, the IIoT Consortium is working on integrating in their standardization also the DT aspects [25]. There are several efforts for standardizing IoT systems [223], [224]. These attempts have applicability, due to the ability of IoT systems to represent the evolution of the context in a specific environment, to the DT. In particular, the Alliance for Internet of Things Innovation (AIOTI) definition of a virtual entity strongly associated with an IoT service [225] is extremely relevant to build a relationship with the DT. At the architectural level this definition is an important step toward interoperability of IoT and DT systems.

Interoperability is possible at different levels, it spans from the sensor levels [226], up to the semantic level [227], [228]. A lot of effort has been put on the standardization of IoT protocols [229], [230]. It has led to a more consolidated choice of a few alternatives [231]. For instance, CoAP [232] and MQTT [233] are largely used depending on applications needs. These protocols can also be reused in DT supporting systems.

The properties of the DT, augmentation and memorization, actually offer the possibility to standardize the APIs of DT, as proposed in [23], and to promote its programmability [234], [235]. All these activities are demonstrating the possibility to program in the large IoT systems, for example, in the context of smart cities [236], [237] or in other situation critical environments. DT will benefit from this interoperability effort.

C. Business Value

With respect to the value offered to customers, the adoption of a DT strategy for products/services is very rich in consequences. First of all, it implies a higher control level and an improved ability to manage the characteristics of a product (on a large scale) in its daily operation. In each moment of a product's life cycle, producers and customers could have a clear idea of its functioning. For instance, it is possible to check how many failures and issues have been encountered and how many of them have been resolved, and how the product has behaved with respect to expectations and possibly to service level agreements. In addition, customers can benefit from additional functionalities and services offered by the DT. For instance, customers interested in the environmental impact can choose configurations and usage of their products with the minimal possible impact, and others, more concerned with other parameters, can choose configurations that better reflect their needs in terms of the usage of the PO. Any services offered in conjunction with a product could have very deep impacts on the users and on potential customers. Some products, such as cars, could create a large ecosystem of products and services associated with the "car object." Users can choose between different offerings on the basis of personalized parameters. For instance, usability, that is, how easy the functionalities are to use and thus to benefit from. Usability and richness in functionalities could have a large role when selecting a product, or a bundle, over others. The products/services will always be updated to the latest version without, or with minimal, user intervention. This has to be as much transparent as possible and avoid a hands-on involvement of the user. On the other side, it also has to be clearly communicated in order to provide the "feeling" of the improvements of the products, that is, similar to what the mobile phone industry is doing. The products will always be available physically or, most of the time, logically. The control on the status of a product can be exerted by the user any time s/he wishes, and accordingly to her/his needs. The product could continually be offered and updated in order to guarantee the maximum reliability. If there are problems, the product itself can trigger its preventive maintenance actions. They can include remote maintenance or request human intervention in such a way as to minimize the user impact. For more innovative solutions, the DT can put in place self-healing and self-configuring capabilities in order to minimize the disruption for the user. In the case of a failure, the product may be modified to a safer status or even be fully recovered. Since products are used in different ways, the DT concept could be very helpful for tuning the performance to each user's current needs. This can increase the customer satisfaction because of personal improvements and optimization related to how a product is actually used by the specific customer. This tuning will inject flexibility into how a product functions, thereby offering improved performances for large set of similar users. In addition, security can be increased and offered to customers by means of updates, improvements and by the tuning of product functionalities.

In general, a single product, by means of the DT approach, could be tailored to the lifestyle and to the current preferences of individual customers. This will very likely result in improved customer satisfaction. It will reinforce their impression of having made the right choice in selecting a specific product.

From the enterprise perspective, the value of a DT implementation includes the following.

1) The ability to continuously monitor and control the functioning of their products from design to operation, up to end of life. This may bring several advantages, such as improvements in new product versions, the identification of weak points and development of counteractions, a better understanding of how a product is used by many different customers, the possibility to intervene before critical failures, as well as a large set of best practices and other aspects about the product.
2) The creation of a continuous link between an enterprise and its customers. For many products, the link

is lost after the purchase of a "good," so that the customer reconnects to the producer only when there are maintenance issues or critical failures. The DT implementation will allow the product to be tailored to specific customers and thus act as a means to collect very specific requirements from different customers. By minimizing the product malfunctioning and failures, the DT's continuous link will provide a better customer experience.

3) The possibility to optimize all the processes related to a product, from design to construction, delivery and operation. The entire life cycle of a product can be followed precisely and improved where needed to provide customers the best possible experience. In addition, when there are customer-to-customer sales, the link can be extended to new customers and security checks can be performed, for example on ownership reallocation.
4) An enterprise can enter the service market in order to augment the capabilities of its product(s), the customer appeal and user experience. In certain cases, new business models can be proposed or implemented to exploit a product's specific capabilities. New revenue streams could then be considered thanks to the flexibility and agility of a product and its softwarization.

The value of the DT concept and its implementation can vary between industries and different scenarios and use cases. It should be emphasized that the DT could represent a very large and complex system without necessarily using the composition's property, for example, a smart city could be a represented as a DT without necessarily requiring that each individual component, for example, buildings, of a city be represented as a DT.

The smart city could then be represented by the set of available characterizing data and it can offer customers a relevant set of information. This means that current IoT solutions could be integrated to provide a view on a large system as a DT. This approach offers the opportunity to integrate and develop new technologies and specific solutions that can scale up over time toward a more granular representation of the single aggregated system. The DT could be seen also as an intuitive way (a metaphor) to present a large and complex system to customers from a more understandable perspective.

D. Toward an Initial Value Chain for DTs

There can be several entities involved in the virtualization effort implied by the DT concept. In fact, different providers could be part of a single DT ecosystem. For instance, PO owners can be seen as the providers of POs that will be virtualized and managed by a virtualization platform provider. The relationship between them should be regulated and managed, possibly, by means of service level agreements in order to provide the basic and extended properties of the DTs to application providers

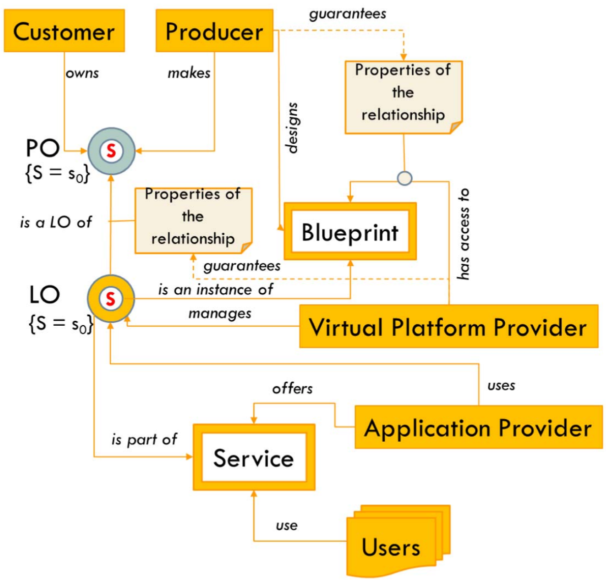

Fig. 10. ***Potential ecosystem for the DT.***

and eventually Users. Obviously, a PO provider and the virtualization platform provider could be the same business entity. The final customer may rent or own the PO, while allowing the virtual platform provider to access the data, communications, storage, and processing capabilities needed to support the DT concept. The relationship between the PO and the customer and the producer is very important for enabling new business models, and for servitization in particular. Fig. 10 represents some possible relationships in an intuitive graphic inspired by UML. The status of the PO and LO is indicated as S, and the initial value is set to s_0 in order to represent the synchronization since the initial stage.

The chart shows how different business roles are distinctive for each actor. In this case, a customer owns the PO, produced by a producer. This actor has granted access to the full description of relevant features of the PO to a virtual platform provider by means of a contract. The virtual platform provider can instantiate the software version of the PO (i.e., the LO) and create a strong relationship with the PO, described by the properties of this relationship that characterizes the DT, and it is the guarantor of this relationship. The virtual platform provider allows the access and the integration of the DT to an application provider that can build a service that will be provided to the final users.

From a value chain perspective, four points of value aggregation can be identified.

1) The POs, that is, the products sold or leased to the final customers.
2) The virtualization platform, that is, the set of functions, data and representations of objects that behave in synergy with the associated POs.
3) The interfaces and views on LOs that can be used in order to create services.

4) The services and applications that use the DT functions. They can be sold or offered to final customers.

Depending on the specific application area and the industry, two points are most likely to embody the greatest value in terms of revenues: POs and the virtualization platform. Services and APIs are instead a means for creating a large ecosystem of customers and users. The value of ICT platforms has been widely studied and analyzed [239]–[243]; the DT infrastructure could potentially become another case of large interoperable platforms that deliver value to multiple stakeholders and actors.

V. SOME APPLICATION SCENARIOS OF DTs IN THE CONTEXT OF IoT

Thus far, this article has addressed the salient features and characteristics that a DT should support, as well as its general technical and business value. It is now important to show why and how to use the DT concept in promising application scenarios. These scenarios were selected to demonstrate how the DT concept can be used in different application domains and are not to be considered exhaustive of all the possibilities. The application of the DT concept is not proposed here as a panacea for all of the possible application domains or for the full implementation of the IoT. Instead, the DT should be implemented when it shows clear advantages over other approaches and when its applications can lead to novel business opportunities or technological breakthroughs (see Section V-E).

The chosen examples are as follows:

1) virtual sensors (IoT);
2) the digital patient (e-health);
3) the digital city;
4) cultural heritage.

These four examples cover a large spectrum of application domains and highlight the flexibility of the concept. In addition, they also target to areas that may have interesting business value.

A. Scenario 1: Virtual Sensors

This scenario considers the possibility of creating an LO representing a sensor. It explores some of the possibilities that such an approach could offer from a technical perspective. A single sensor is considered first, and then a collection of them is used to provide an overview of a possible solution.

The case of a single sensor offers two different implementation options:

1) a single instance of the LO associated with the physical one to form the DT; or
2) several LOs associated with the physical one in order to provide a specialized logical entity to the different requesting applications.

Each application, in fact, could have specific requirements: one may request access to the sensor values within stringent time intervals, another could request the values by polling the object, and so on. The LO is associated with an application by means of a specialized relationship represented by APIs. There are also different deployment options that may be implemented: for example, deployment and instantiation of the LOs only in the virtualization space and supporting platform, or in the domains of the application providers.

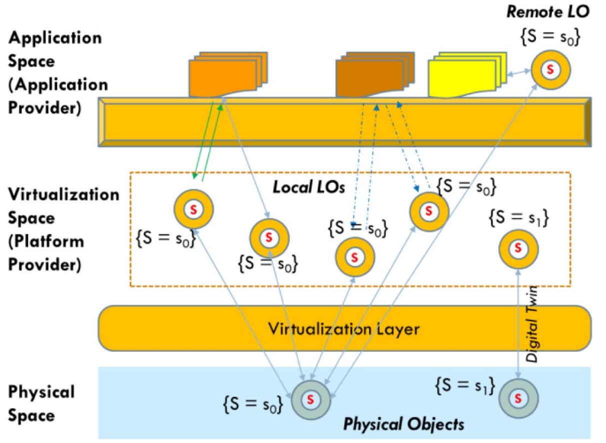

Fig. 11. *Virtualization and replication of LOs.*

Fig. 11 depicts a PO in this case a sensor, mirrored and entangled with different software replicas, that is, the LOs. Each replica is instantiated to offer information and functionalities to a limited set of applications. It can fully satisfy the specific applications requirements in terms of collection of data or management policies. If the PO is programmable, the LOs can execute commands on the physical device, in this case, an actuator, for a specific application case. A few possible issues should be noted. The most common one is that the PO will often have scarce processing, storage and communication capabilities. Increasing the number of related LOs, that is, objects PO will have to interact with directly, may require levels of processing, communication and storage power that could easily and quickly exceed the PO's capabilities. In addition, different communication links can be difficult and costly to maintain. Moreover, different LOs should be strongly synchronized with the PO maintaining its consistency, for example, no conflicting commands or changing of status. This requires the continuous and effective updating of all the involved objects.

The different LOs do not necessarily have to be synchronized amongst themselves, but they should be in synch with the PO. The need for synchronization may lead to a situation where the status of the different replicas is constantly updated even if each LO substantially represents the same state. If strong consistency between the status of all the LOs and the PO is needed, a transactional system to synchronize all the objects should be introduced. However, it may result cumbersome and, in some cases, not useful if the LOs can be independently updated. A Publish Subscribe (PubSub) system [244] can be used

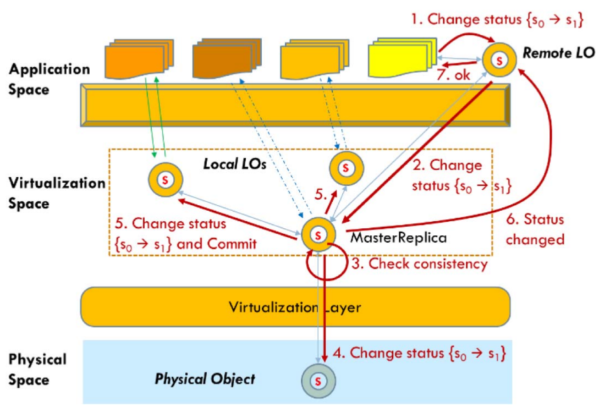

Fig. 12. ***Controlling the different software replicas, LOs, associated with a PO in a DT.***

to simplify the communication between the PO, a publisher of information, and the replicas, subscribers to some information.

A specific case to be considered is when the PO accepts commands and can change its status accordingly. This capability is subsumed by the strong entanglement property. The applications, then, can individually request a change of state in the PO. This situation has to be well-orchestrated and managed in order to avoid malfunctioning and inconsistency in the PO, and thus this case requires a synchronization mechanism. Fig. 12 depicts a simple case in which a MasterReplica governs, in a centralized manner, the change of states of all the entities. A MasterReplica is an LO that supports the synchronization between the PO and all the other LOs. A transactional system could be implemented to synchronize the change of states of all the LOs. Change of state requests from specific applications are passed on to the MasterReplica that will manage them and mirror the changes to all the other replicas in order to maintain consistency between all the LO instances. When the MasterReplica receives requests to change status, it checks if they are consistent, communicates, and controls the change of the PO, prepares the other replicas for the change and then commits the system to the change.

Other forms of consistency enforcement can be implemented according to the needs of the different applications. It is important to consider that in these circumstances, that is, the distribution of replicas, the requests for the quick availability of data, and their consistency, the CAP/Brewer Theorem holds [245]. This theorem could greatly affect the behavior and the performance of large DT distributed systems. However, from a practical perspective, consistency and availability can be achieved within a reasonable and practical timeframe [246], [247]. Consistency and availability of large distributed systems can be achieved under specific usage conditions and requirements. DT platforms can exploit these characteristics and solutions for a large part of their potential applications.

In cases where more than one sensor is virtualized, the aggregated set of sensors can be used by one or more applications. Some relationships between the sensors can be determined over time by the virtualization platform, for example, by means of ML techniques, and exploited to improve the functioning of the sensors. Fig. 13 shows this particular case.

It is assumed that the virtualization space, and the supporting platform, is capable of managing the functioning of the LOs. To this end, an Orchestrator can be used. It is also capable of analyzing the behavior of the monitored LOs in order to detect their malfunctioning as well as their reciprocal relationships (by using continuous data analysis). For instance, all the physical sensor values are captured and stored in the DT system in order to be available for analysis or for system restoration. The collected data can be analyzed to determine the relationships and links between different sensors, for example, temperature sensors. In this case, two sensors are strongly related to a third in such a way that its value can be inferred by the values of the other two. An orchestrator can then instruct the LO of the third sensor to put the physical sensor "on hold" while continuing to provide information (inferred by the other two sensors) to the requesting applications. Any now and then, the third sensor could be woken up in order to check the alignment between the data computed by the LO and the actual data measured by the physical sensor.

This example has illustrated some of the constituent properties of the DT. Mirroring, reflection, virtualization, strong entanglement, and memorization were used to describe the expected functionalities. It is increasingly evident that the implementation of a DT infrastructure requires the identification, design, and implementation of a rich software architecture in order to fulfill the requirements and support the expected properties. In the following examples, the need for a proper architecture will emerge even more clearly together with some of its important basic system functionalities.

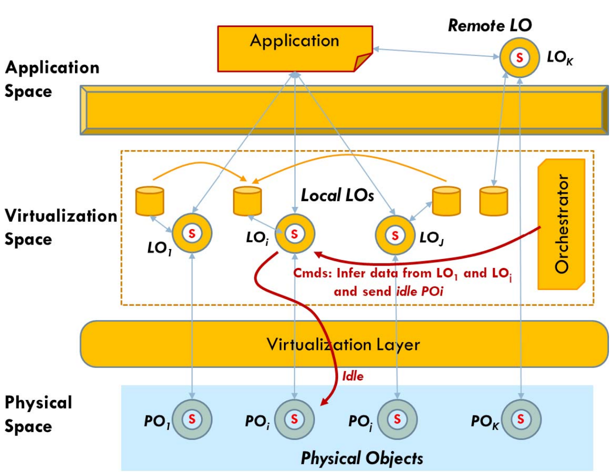

Fig. 13. ***Smart orchestration of DTs.***

B. Scenario 2: The Digital Patient

One of the major issues, with relevant costs, associated with health care is the monitoring and observation of patients. It comprises the continuous and accurate collection and storage of patient data. They will contribute to create and update a complete medical record reporting the information about the patient. This record presents current and historical parameters. It should, possibly, comprise reliable information on the patient's lifestyle. Standards and initiatives that seek to consolidate the best practices and provide a complete set of medical information for patients are being implemented [248]–[250].

With the advent of the IoT and the possibility of fully monitoring clinical patients with a wealth of body and environmental sensors and low-cost devices [251], there is the possibility to implement many different solutions to improve patients' quality of life without requiring their hospitalization. The availability of sensor technologies as well as the growing communication and embedded processing power opens the path to the application of the DT concept in healthcare [207]–[209]. In fact, the DT can represent the "Digital Patients" in their own context, for example, at home, during daily activities, etc., offering very effective ways to monitor and interact with the real patient. A digital patient would thus be an implementation of the DT concept fully devoted to monitor and represent the status of a human being. The goal is to provide a better patient care by offering a comfortable real-life context out of a hospital. Physiological data, actions, interactions with the environment, activities, and other parameters (e.g., psychological expressions such as laughter or the sound of their voice) could be collected and used to identify both the risks and the positive activities that a specific patient may be experiencing. The digital patient would be a very tailored set of monitoring capabilities. It may comprise sensors on the patient's body, sensors in the environment, including in the usual spaces s/he lives in. They are useful to understand specific description of limitations and pathologies and to associate them with related measurements on the field, as well as a set of alarms and warnings to be issued to the patient and to the controlling medical team.

The digital patient is also posing very stringent requirements in terms of security and privacy. The DT implementations in this sector should guarantee an extremely high level of protection for the personal data. The digital patient could also be useful for the implementation of the Virtual Patient framework [252], a software framework that simulates real-life pathologies in such a realistic way that practitioners and students can exercise and improve their skills at dealing with different diseases. This feature could be a byproduct of the implementation of the digital patient. Very detailed data about different real-life patients could be collected, analyzed and used to create effective models of "Virtual Patients" for training or for research. In addition, data collected from several specific patients could be organized, analyzed, and compared in order to determine a general pattern in the evolution of some diseases as well as variants of the diseases. AI techniques and the ability to acquire data from several digital patients could play a major role in the prevention and understanding of maladies. Two major types of patients could be targeted for the digital patient.

1) *Monitored:* Those that must be physically monitored by means of specific devices (sensors, Holter, and the like). Some may be monitored for a limited amount of time, others for longer periods.
2) *Nonmonitored:* Patients that have a pathology but do not require constant monitoring. They only have to be checked for general parameters on a sporadic schedule.

The first set should be continuously monitored and hence a long lasting and secure association between the patient and its DT is important: the strong entanglement property of the DT could be very effective for these types of monitoring. These patients are most likely under surveillance from doctors. Their medical records are often updated and checked daily. The goal is to determine significant variations in vital measures and to timely report the medical team. The second set of patients can be associated with a weaker entanglement and hence data may be collected and sometimes crawled from several sources without restricting the patient to a specific physical space fully under control.

In both cases, the collection of data referring to the patient has a paramount importance. On the one side, the medical records comprising current and historical data, contextual data derived from the life habits and circumstances of the patient's life are updated. The one with pattern of illness or disease evolution inferred from the literature and from comparison with large sets of other patients data is also considered through data analysis. These two data sources contribute to the detection of issues and their prevention by means of a continuous monitoring of the DT.

The DT is a means to search, access, collect, and store and continuously relate and evaluate several sources of relevant data of the patient. In fact, each DT could retrieve and store specific data of the associated patient: such as patient's analysis, information on the patient's lifestyle, circumstances of particular importance. For instance, exposure to particular substances present in an area or the fact of living in an area that is more polluted than others may have an impact on health. The DT could collect a wealth of data related to the typical food intake of the patient, his habits (sport activities, sedentary life, smoking habits, and the like). All these data are collected through the explicit permission and cooperation of the patient, through actual measurements, or taken from the Internet by accessing to data of the most used applications of the user). For certain diseases also psychological status can be relevant: the DT can crawl the data in social media or

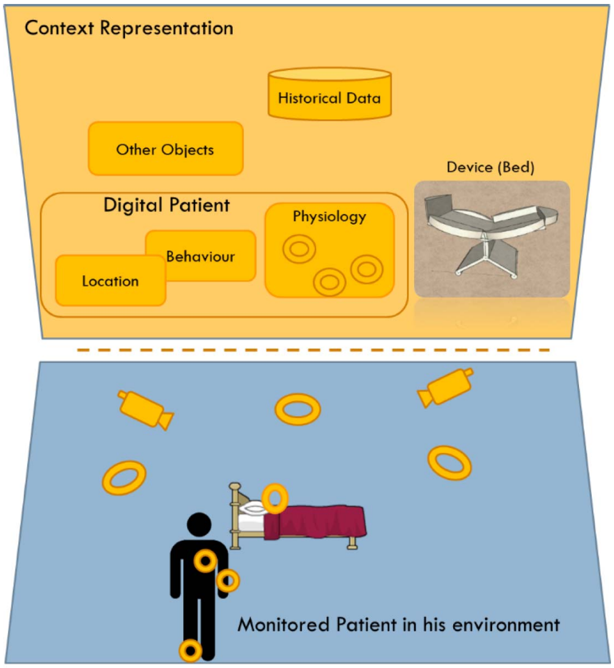

Fig. 14. ***Digital patient scenario.***

communication applications generated by the patient in order to determine its behavior or feelings. The DT then will be able to support the storage and the access of the patient's data and will support the evaluation of patient's status, and his contextual information. These activities may occur under the supervision of medical personnel. The DT could also collect and compare in an anonymized fashion these data with those of other patients. This will permit to determine patterns in the evolution of specific illness, or determine thresholds of risks. The collection of data will contribute to the identification of best practices. For instance, DT can help in determining when the specific patient has to undergone surgery or must have specific treatments. The DT can also emit warnings or alarms in critical or important situations, or even to predict the possibility that a critical situation can occur. In addition, access and analysis, if available, of genomic data of a specific patient could bring to very useful information about how a specific disease can develop or can be prevented, or how a disease can impact on other peculiar physiological aspects of a specific patient. The digital patient can store, update, and analyze this genomic track record. If this approach is applied to a vast population and it is contextualized in specific areas and/or associated with specific lifestyles, this could lead to unprecedent information about how to treat a large part of the population or how to intervene for specific polluted areas.

Fig. 14 represents the case of a patient in an instrumented context for patient monitoring.

It comprises also other programmable objects devoted to make the patient life easier and more comfortable. The DT application leads to the definition of a precise portrayal of the patient derived from different information sets: the physiology data collected from body sensors, location data inferred by motion sensors, and action and behavior data extracted and inferred by analyzing images by cameras. These sets of information and data originated by different POs are aggregated and they form the basis for representing the current digital patient status, in that specific context. In addition, other programmable objects, or DTs, can be considered. In Fig. 14, a special bed is considered. It can be positioned in such a way to mitigate the pathology of the patient, that is, for example, particularly prone to postural issues.

From a functional perspective (Fig. 15), the constituents of the digital patient will be the digital representations of body constituent or characteristics of the patient in terms of physiological, location, and behavioral parameters plus other DTs that do represent and are able to control the context in which the patient is operating.

The DT implementation has to be supported by functionalities that are related to the management of several sources of data and their fusion, the aggregation and composition of different LOs in order to represent the Patient and his/her capabilities, a set of cognitive functions in order to "understand" the environment and to predict the patient's needs in order to accommodate the context to them. In addition, applications will control and coordinate the needed functionalities according to a precise set of medical goals and strategies. These functions are an important part of the functionalities and services that a DT platform has to provide as general features.

The simulation capabilities of the DT could also be interesting from a healthcare application domain perspective. It could help in proving and determining models to predict the evolution of diseases. For instance, if the initial hotbeds of influential are known, then some models based on the mobility patterns of people could predict how the disease will spread and how to better cope with it. Other functionalities could be considered and added to the platform feature set. The patient could be fully supported in well-known environments. In case of bedsores or back prob-

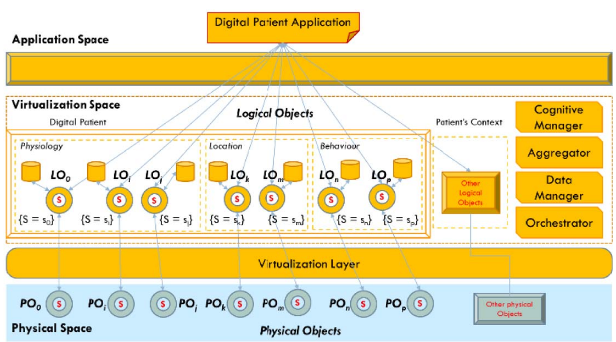

Fig. 15. ***Functional architecture of the digital patient.***

lems, the bedding system could be controlled in such a way to minimize the damage by positioning the patient in the best possible way. The food intake could be controlled by sensors in the fridge, in the microwave oven, and by cameras in order to provide the best possible diet. The timely assumption of medicines and remedies could be controlled and regulated accordingly to the real needs of the patient, for example, under skin installed dispensers of drugs associated with the pathology of the patient and controlled by the DT in order to guarantee the exact dose intake.

In this case, the basic properties of the DT concept are: reflection, virtualization, and entanglement. They are to be offered in a very secure and constrained manner because it would be critical to lose contact with a patient with serious diseases. In addition, the composability, contextualization, and the memorization properties are very important. Composability means that different objects, for example, devices for the heartbeat, pressure, and the like, should be composed into an aggregated object that represents the physiological status of the patient. The context in which the patient is operating has to be understood and possibly controlled in order to make it easier for the patient to move and live in the surrounding environment. The data collection is extremely important in order to detect critical parameters and emitting alarms as well as for identifying patterns and to compare them with those of other patients. Comparison and evaluation contribute to identify the best practices for taking care of them. Augmentation is also extremely important because it offers the possibility to doctors or other personnel to check on demand some body values or to accommodate the environment to a particular need that could emerge.

C. Scenario 3: The Digital City

The concept of DT has been frequently associated with the smart city studies [138], [175], [196], [253]. In this case, the DT is very intriguing because of its properties of composability, memorization, representativeness and contextualization, augmentation and servitization. In simple terms, adopting a DT representation for simple objects up to a large aggregation of them, for example, a hospital as the aggregation of different compound objects, can provide several abilities to govern the city. Some possibilities are: continuous monitoring, programmability, big data analysis, services and applications offered to citizen and other actors, predictability of certain phenomena like traffic or congestions, and many more. The DT concept is well aligned with the idea of smart cities as complex systems [254] that show real-time behavior as well as emerging ones in the long run. The properties of the DT seem to fit in the need to control small contexts and environments and to be able to scale up to understanding and controlling the macro-behavior of the entire city. It is evident that modeling a city in all its aspects by means of DTs is a daunting endeavor. More practically, the approach is to apply the DT to small parts of it and to scale up over time. Actually, the DT concept could be adopted also at higher levels, for instance by representing a large and trafficked avenue, or an entire neighborhood. In this sense, a bottom up approach, from small objects and sensors up to large aggregations of objects, and/or a top down approach, from large systems to their decomposition into smaller ones, each one representing an important city object, could be possible and implementable.

In the case of the digital city, it is also important to note that the DTs, by means of augmentation and servitization capabilities are the basic for open solutions that do not create siloed and closed vertical application domains. So, an object representing a crossroad can be used for logistic based applications, for traffic control as well as for security. In addition, the memorization property plays a fundamental role in terms of collecting the data and helping in determining emerging patterns in the city behavior. This property, associated with the predictability one, is very important for predicting the behavior of the city. If detailed measurements of its characteristics are recorded and analyzed, for example, the impacts of the introduction of a new "one way" road, new traffic lights, new buildings, and so on could be simulated and studied before to decide their implementation. Another aspect that has to be considered is that a smart city could be related to DTs representing other large environments. For instance, in many countries there are "twin" city, that is, cities that have a deep tie and they strongly interact even if they may be far away. These twin cities could be related and studied in order to better determine the effects of events occurring in one city over the other one. The DT in this case can be useful for analysis at the higher level as well as for very specific relations, for example, transportation.

Typically, the IoT and the smart cities applications suffer from the issue of applications silos, that is, the strong separation of architectures, tools, interfaces, and data between different applications domains, transportation, traffic, cultural heritage, and more. The DT concept can somehow help for a better composition and aggregation of functionalities independently of the specific application domain taken into consideration. The reason for this is related to the bottom up approach implied by the DT. Each single and simple object can be accessed by APIs and it can be composed into several aggregations of other more complex objects. Each object can be replicated at will and it could be specialized for the particular goal of the application. Still it will be entangled and synchronized with the original and possibly with other instances of the object. This ensures a high level of composability and the programmability. In order to leverage these programmability and composability features, an important architectural function within a DT platform should be considered: the brokering of available objects. Fig. 16 shows schematically a few phases for introducing a new PO and the creation of the DT relationship.

The owner of the PO requests the association of "its" object to the DT platform. The Brokering Manager will deal

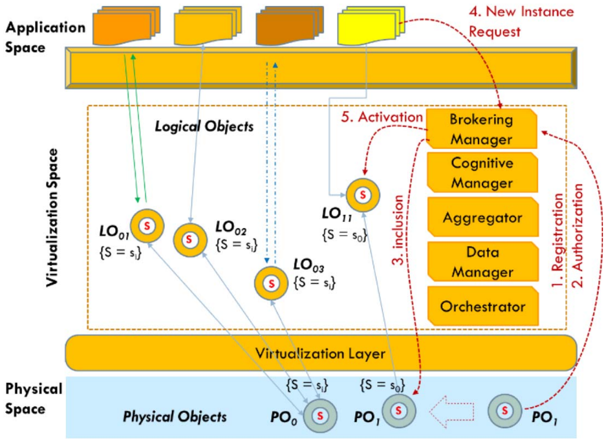

Fig. 16. ***DT brokerage.***

with the request by checking the registration parameters, by authorizing the object to access to the platform, and adding it to the list of available resources. The inclusion phase means that a relationship between the PO and a new LO is created and the system is able to provide and support the expected DT properties. The activation phase takes place when the DT is actually instantiated and executed on the platform. It can be subsequent to a direct request of an application, as shown in Fig. 16, or it can occur as soon as all the checks and the verifications have been executed and the DT is able to operate. That means that one or more LOs are associated with the PO and the flow of data is actually collected. In this case, the presence of a Master LO could help in the development of new applications requiring the instantiation of personalized LO.

One strong requirement of a smart city platform and its applications is the ability to deal with a multitude of different objects and even more with plenty of their instantiations and replications. In the future, the smart city applications could scale up to hundreds of thousands or millions of objects to be controlled, governed, and orchestrated. This requires that the platform is capable of providing functionalities for self-management. It will not be possible to manage and orchestrate a multitude of objects by human intervention and configuration. Self-organization is a fundamental function to be guarantee in these kinds of systems. Fig. 17 depicts the situation in which one object detects some malfunctioning and it is able to warn the system of the issue.

The self-management function synchronizes with the Orchestrator in order to determine where and which storage, computing, and communications capabilities are required and where they can be allocated. Then a new instance, that is, an LO, is created, a status value is assigned on the basis of the PO and other LOs, and the application is bond to it. A synchronization phase should be carried out in order to limit the impact on the application. Ideally, the application should not be aware (and in any case not be affected) of the change in the instance of the LO being used. In this specific case the self-management functionalities are shown as a functional block. It depends on the different implementations of the DT platform to opt for a centralized or for more distributed solutions.

From the smart city application domain, the interesting properties of the DT are related to the following.

1) Representativeness and contextualization in space and time, because certain objects, for example, a traffic light system, a crossing, a building, can have specific characteristics and behavior that can have an impact on other city activities.
2) Composability is important for the ability to aggregate simpler objects in more complex ones and to scale up in size and still have control on components and the whole aggregation.
3) Memorization; in order to study the behavior of a city or part of it. It is fundamental to collect historical data and to be able to analyze them in order to determine patterns and issues and to predict future behavior.
4) Augmentation; the capabilities to add functionalities and to program the DT are salient abilities for impacting on the behavior of the city and to better adapt to the current situation or to try to modify it by intervening on crucial objects.

D. Scenario 4: The Cultural Heritage Scenario

The DT concept could also be applied to the representation of cultural artifacts. A painting, a statue, a watch, a work of art can be represented as a DT. These kinds of POs are not simply objects, they have physical characteristics and features, but they also have a relevant content, an idea, associated with them. They are relevant for the cultural information they carry or that has been associated with them. For instance, the Bernini's Bust of Louis XIV in

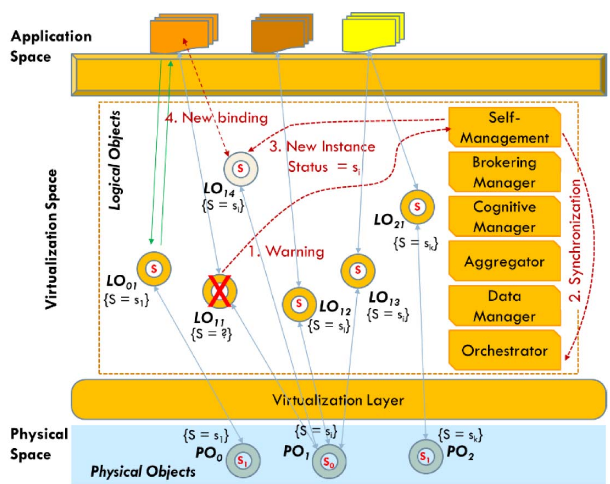

Fig. 17. ***Self-management and healing in a DT system.***

Versailles is considered the grandest piece of sculpture of the baroque age. Behind the object there are its representation, the context of the time, the techniques for realizing it, the functionalities it intended to offer, the innovation it carried, the appreciation of people, or their criticisms, over the years and centuries, the representation of a style and concepts. In addition, the ownership is important to determine the historical relevance of the artifact. All this relevant information has to be captured and represented to the applications using the DT of these artifacts. The point is crucial because they are real-world objects and they have physical properties, but they also could represent advancement in how materials were used, or strong implications with beliefs or way of thinking of different ages. These objects are then to be viewed under different facets.

1) A physical view, how the object is made, what components, materials and what is the current status and location. This information can also consider the data collected by sensors in the environment: humidity, temperature, or security information, as well as location and others. This information could also span over different periods of time during which the artifact was moved, its owner changed and the like. The current data are relevant for preservation of the artifact, while the historical data are relevant because they describe the "physical life" of the artifact.
2) A functional view, that is, what the intended use of the object is (e.g., a watch, a statue, and a building) and what functions it provides or was providing. Also in this case, the historical data can describe how the usage of the PO has changed over a period of time.
3) A cultural view; that is, how the object has been perceived, interpreted, studied, and appreciated over the periods. Certain objects have gone through periods in which they were neglected and period in which they were considered as very important. These changes in evaluation depend on historical periods and mutation on the "sentiment" of people and particular moment in the history. Collecting these data is very difficult, but it is fundamental in order to fully understand the PO.

Cultural objects are not only determined by their "physicality" but mainly by the context and the perspective of people over time. There is the need to access to relevant cultural information associated with the "Objet d'art." Fig. 18 depicts a situation in which data are collected from different sources: the entanglement with the PO, data offered by Curators, and data retrieved by means of crowdsensing from visitors and users.

The cultural heritage scenario is useful for exploiting the definition of the Ownership property. For instance, a replica of an artifact can be offered to a user, the ownership of the replica, and its rights and manipulation possibilities with respect to the original, should be clearly represented and traded in order to avoid conflicts. In addition, from an historical perspective, a story of the changes of the ownership of an artifact are important to determine the cultural context in which the artifact has been produced and also the perception of its value along the time.

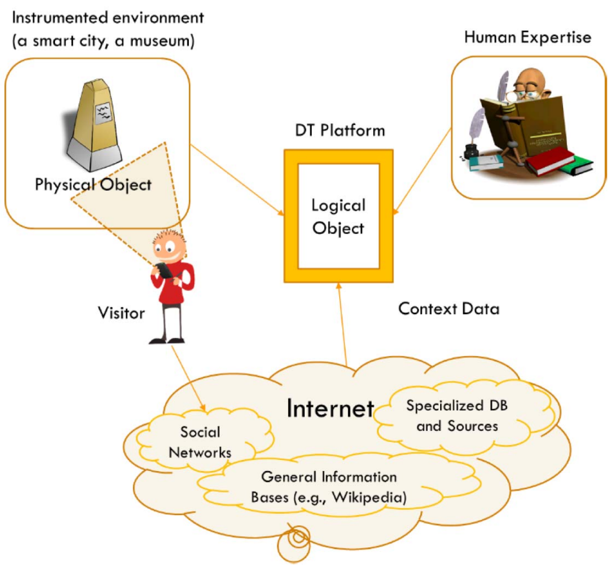

Fig. 18. ***Sources of information for an "objet d'art."***

For a deeper knowledge about the cultural value of the artifact, there is the need to access to other relevant information or the support of curators and experts of the sector, that is, humans. However, a wealth of information could even be inferred by the continuous access and crawling of information available on the Internet like general wikis, specialized data bases, or even information extracted from social media and networks. In this case, the DT acts as an agent that actively seeks information about its physical counterpart. Additional sources of information about an artifact are depicted in Fig. 19.

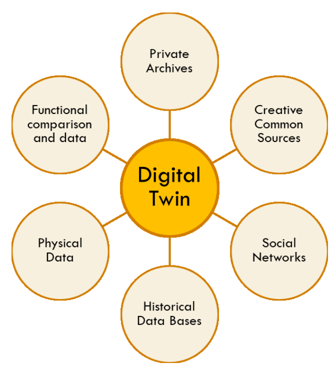

Fig. 19. ***Crawling information about an objet d'art.***

The DT implementation has the goal to manage this data and to proactively crawl different sources for new information. The different DTs will also behave in such a way to create relationships between them in order to identify common properties, for example, the same subject, or the same owner, or the same usage of colors or other techniques. This search for relationships will create a network that could be evaluated by historians and experts in order to figure out new associations between different artifacts. The DT could be an additional tool for seeking relevant historical information.

The cultural heritage scenario pinpoints to an extremely complex problem: the capture of historical and contextual information that span from physical to cultural realms and its organization into semantic and reasoning systems for information capture, extraction, and manipulation. The DT can use its capabilities and properties in order to create a continuous entanglement and search for information related to itself and its context.

From a software architecture perspective, the cognitive aspects are extremely relevant as well as the continuous information crawling, formatting, and assessment. They can fully complement the idea of the entanglement with the PO and the passing of status information. Fig. 20 is based on the previous architectural model. It shows additional functionalities and represents a sort of recap of the capabilities considered so far for the different applications scenarios. In the case of cultural heritage, the data management part and the cognitive and AI mechanisms constitute two fundamental features to support the concept in this relevant application domain. They are the needed functionalities that allow to cope with the cultural aspects and the extraction of new interpretation and "understanding" of the artifact. They are an integral part of the architecture and they well conjugate with all the other functionalities. The DT is a step further respect to the digitalization of cultural heritage. A DT can graphically represent an artifact by means of augmented or virtual reality technologies, but it offers much more. A DT is actually a softwarization example of artifacts.

Fig. 20 represents a large architecture based on middleware functionalities that are deployed over different computing systems like terminals, edge and cloud. It also provides the possibility to program and to extend the functions by means of external and internal APIs.

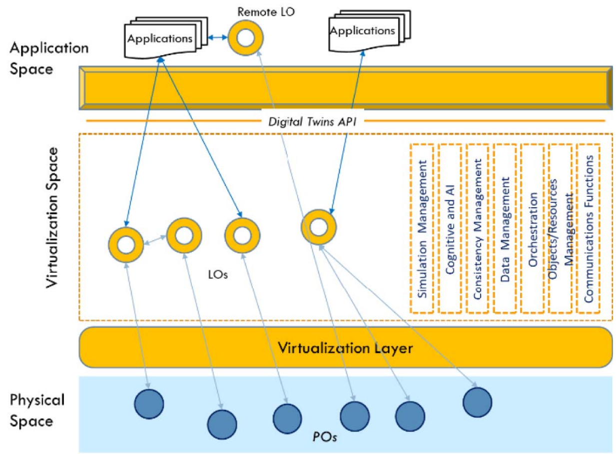

Fig. 20. ***Architectural model for the DT.***

E. When to Use a DT

DT is finding interest and an increasing application in several fields. The advantages of the DT are several which are as follows.

1) Its ability to represent POs, in fact LOs are models of the POs.
2) Its capability of collecting and representing large set of values and attributes describing the PO.
3) Its ability to place in time and space the PO.
4) Its ability to relate with other DTs to investigate and figure out relationships and communalities between objects.
5) Its ability to serve the life cycle from creation to the dismissal of POs.
6) Its ability to support servitization.

These capabilities make a DT applicable to a large variety of scenarios and application cases. However, the DT also introduces a level of complexity in terms of modeling of POs and environments. Entanglement may not be possible for specific objects, or usages and applications. The aggregation of DTs into a larger aggregation may be difficult, and it may require a lot of processing. Not all the system will require a large historical data set, or alternatively, these data can be stored as simple files, for example, in comma separated value (CVS) format. Simple applications do not require servitization capabilities and complex level of management of ownership.

The current offering of IoT applications may cover a large part of the needs and requirement of use cases. Also AI applications, if data sets are available, could be competitors of DT implementations.

The value of the DT usage has to find a correct balance between the complexity of its application and the possibility to introduce new business approaches. Under this perspective, large and cross-domain applications could benefit from the capabilities of the DT for accurately representing large aggregation of objects and to transform them in services.

From an IT development perspective, there are some solutions that are maturing. They are derived from the work done for IoT and they are extending their functionalities in order to support the DT. For instance, an architecture as FIWARE [256] is a good platform moving from the research into a consolidation phase for wide commercialization. It is able to support DT applications [212] and providing some basic functionalities needed in a DT platform as identity management [257], cloud resources management [258], and semantic approaches for data analysis [259]. In particular the ability of dealing with large amount of data through brokering [260], [261] is an important enabler for the DT.

VI. ARCHITECTURES FOR SUPPORTING THE DT CONCEPT

Section V depicted a set of scenarios and showed a general functional architectural model to support the examples. This architectural model can be used as an outline or a high-level design for a DT platform. Actually, academia and industry have already started several specification and prototyping activities in order to demonstrate, develop, and introduce platforms supporting the DT concept. Many well-known companies are involved in positioning their solutions or future products as leaders of the sector. It is difficult to evaluate the real implementation status and the availability of these solutions and their readiness for entering into the market. Some technical trends can, however, be identified in this platform development effort and compared to the blueprint architecture described in Fig. 20 in order to identify for the DT platform some commonalities and needed functions. A first loose differentiator factor in the design of the platform may refer to the background of originating companies and research group. There are some that are actively working with a sound background in manufacturing and, as a consequence, Industry 4.0 or the so-called industrial Internet is their target. These groups base their developments on the requirements and goals of large manufacturing companies that want to implement the DTs concept in manufacturing. For instance, [255], [257], and [263] clearly point to the needs of the manufacturing industry and how the DT and their proposed architecture can help.

There are also relevant attempts to move this concept into practice in the Industry 4.0 [264] or to standardize supporting interoperability solutions for the Industrial Internet.[2] This trend is also strongly concerned with the usage of data in the context of industrial environments [265] and the integration of the platform and the DT solution within the IT infrastructure of the manufacturers [266]. Fig. 21 depicts a current effort of a large industrial platform supporting the DT concepts. Many of the discussed functionalities presented in Section V are present.

An interesting proposal, stressing out relevant properties of the DT like simulation and prediction, as well as collection of historical data in a context of advanced manufacturing comes from [199]. The need for creating mathematical models for the description of objects is, instead, described in [268]. Microsoft aims at being a provider of IoT solutions for the Industrial Internet and the DT is a

[2]For instance, the Industrial Internet Consortium has created a "Digital Twin Interoperability Task Group responsible for comprehensibly defining Digital Twin characteristics with a specific focus on Digital Twin interoperability for industrial systems" see https://www.iiconsortium.org/wc-technology.htm

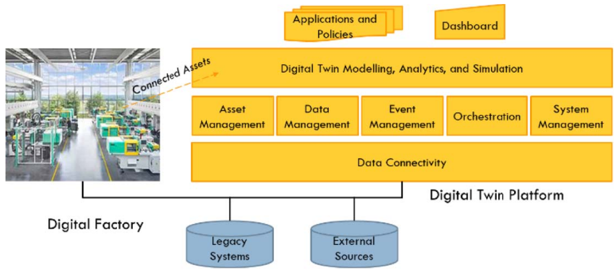

Fig. 21. ***Model of an industrial architecture supporting the DT derived from PREDIX [267].***

concept to be supported in their software infrastructure. The proposed architecture [269], [270] is based on the linkage of devices to the basic platform functionalities like cognitive services and business intelligence, simulation and visualization, and enterprise intelligence and system integration. They also introduce a programmable layer called DT Services that comprises services, applications and tools that can be used and exploited by customers. Also IBM is operating in this context and its proposition is to leverage the Watson IoT platform together with cognitive capabilities in order to model the DTs and providing valuable functionalities to customers [271].

On the other side, the groups working more on IoT are taking their platforms as a reference for supporting the DT concept. One interesting article, introducing schemas for modeling objects is [272]. Modeling of object is clearly a need that emerges in several implementation and platforms. On the industrial level, SAP is trying to leverage its Leonardo platform in order to implement a DT solution [273]. This is offering interesting functionalities and points to be considered by different architectures: Twin-to-device integration (e.g., mirroring and entanglement), twin-to-twin integration (composability and augmentation) and twin-to-system-of-record integration and twin-to-system-of-intelligence integration (i.e., memorization and data analysis). In addition, SAP points to the objects modeling, Thing Modeler, in order to fully describe the POs' characteristics. Amazon is exploiting its cloud and IoT infrastructure and is also putting forward the possibility to integrate the DT concept with some forms of Virtual Reality [274] for a better visualization. Schroeder *et al.* [34] follow a similar path but with a specific focus on Web services architectures.

Augmentation by means of programmability and APIs definition is an aspect taken care by all the middleware proposals, it is worth to point out to a specific API defined by Eclipse that is expressly addressing the DT: the Eclipse Ditto API [275]. It is used in conjunction with other open sources components in order to create a viable IoT-based platform with cognitive capabilities [276].

All these platforms, especially the industrial ones, pay attention to the edge computing aspects and they support the possibility to execute and exploit the edge capabilities in order to better implement the DT concept, for example, [277] and [278] with examples of smart city and DTs. Virtualization techniques are widely used, actually in both paths, and the recent technologies in this field can be useful for segmenting the different functionalities needed to support the DT. Examples of this approach with a specific reference to microservices are found in [30] and [279].

Another important aspect that emerges in the middleware development effort is the usage of functional layers as a means for organizing the platform capabilities. Layering helps with different interfaces and functions to better serve and support the applications' needs as discussed in [280] and [281]. One of these layers is commonly associate to "cognition" or the capability to "reason" about the data and the states of objects and their behaviors. Also, autonomics and self-management are well considered properties of the DTs [189]. From a software platform perspective, the broad implementation and span of the DT concept will require an extreme flexibility in deploying, executing, and managing the software. Functions and services as well LOs and their logic should be promptly deployed within system capable of supporting their requirements. In addition, objects could require mobility and fast replication deployment within a flexible execution environment.

Many general functionalities should also be available on demand and with real time capabilities. Data should be elaborated in almost real time in order to support the logic of the different applications. However, the raw data should not be lost, because they are the major source for determining patterns and pre- or partial elaboration could impact the pattern emergence. Batch processing should also be enabled in order to extract more information. The DT platform and the associated devices will operate and use heterogeneous operating systems and languages. The platform as a whole should be capable of executing different types of software without prescribing to programmers specific languages, operating systems or tools or mechanisms. Software should be "thrown" into the DT platform and be easily executed without requiring the allocation of specialized resources. The paradigms of Lambda computing [282], [283] and liquid software [284] are examples of the requested flexibility. This is also useful in order to avoid a kind of siloing effect due to the technologies used.

Fig. 22 shows a detailed representation of the architectural model previously presented. It considers many of the identified capabilities and functions as proposed by academia and industry. A set of layering principles and functionalities stemming from the architectural analysis of some of the existing proposals is represented.

Layering is consistent with the general trends in large middleware development and it is consistent with the broadness in scope of a DT platform and its needs in terms of separation of concerns. The bottom layers are those that interact with the devices, the edge and the cloud resources,

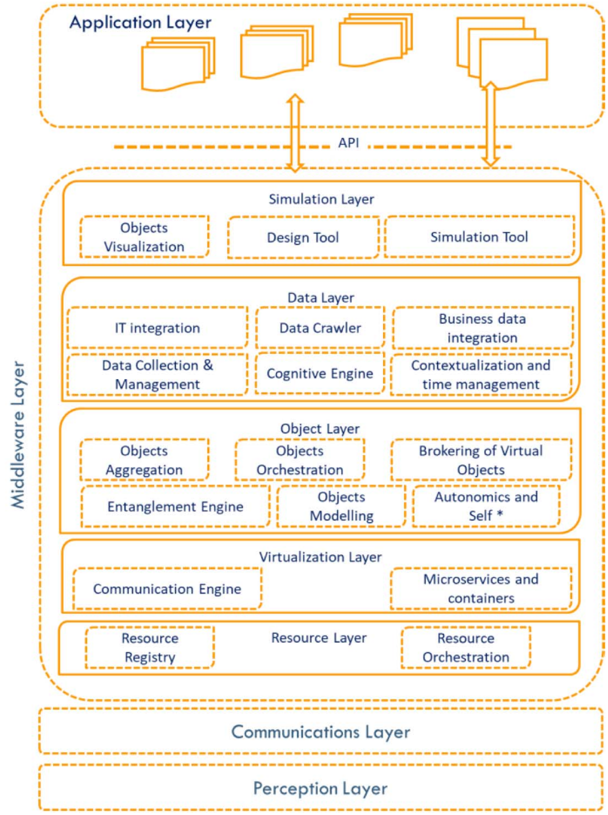

Fig. 22. *General framework for the DT.*

and take into consideration how to use them, how to virtualize and how to exploit their characteristics. Some functions are needed to allocate the right resources and to virtualize functions/objects in the infrastructure. In addition, due to the stringent needs of some LOs, the communication has to be optimized, especially for efficiently supporting the entanglement. The layering is essentially derived from the current developments in virtualization of network resources as defined, for example, by ETSI [285]. The upper layers deal with the major properties of the DT, the object layer is devoted to the life cycle of LOs and their existence. This comprises functions for modeling of objects, instantiation, self-management, orchestration, entanglement, and other. The data layer deals with collecting and contextualizing the data as well as to execute data analysis and information inferring. In addition, there are crawling functions in order to collect external information not directly provided by the POs. Semantic and ontologies could be added in here for specific problem domains. The Simulation Layer supports the visualization of the DTs, their simulation as well as tools for the design and the definition of them.

All this infrastructure is based on open APIs in such a way to be programmable at different level and with different abstraction capabilities. Applications will be able to interact with all the needed platform functions by means of well-formed APIs and structured data.

VII. PATH AHEAD

Sections II–VI have addressed technologies, scenarios, application cases, and architectural models that are relevant for the implementation of the DT concept. There are several others not mentioned in here, and the future will bring more. The variety of relevant scenarios shows that the DT concept is already well accepted by both the academic and industrial environments.

It is important to understand that, to date, the application domain is dominated by prototypes and/or by proprietary solutions, sometimes not fully implemented. In addition, some of them only address specific application domains and lack the needed generality and openness to be widely used. A definition of open standards is required in order to overcome this lack. Already some standard development organizations (SDOs) are tackling this issue, for instance, ISO with ISO/AWI 23247, IIC [286], ETSI, and European projects extending the capabilities of the oneM2M platform, for example, [287], ITU-T SG 20 [288]. A coordinated effort in this sector is clearly needed in order to coordinate the different stakeholders and grasp the relevant requirements, as well as to determine a shared definition of a DT and its properties, capabilities, and interfaces.

From a technical perspective, there are still some major issues to resolve and to prove, for example, the entanglement capabilities, the scaling up of the DT platform to millions of objects, the aggregation capability, the possibility of self-management and the zero-Touch approach, the collection and analysis of captured data, the ability to contextualize DTs, the crawling and the enrichment of related data from sources other than the PO itself, the ability to simulate and predict the behavior of large systems of DTs, and many more. Clearly, there is still need for academic and industrial research in this sector, but one issue is of paramount importance: security. What could possibly happen if somebody is able to hack into or to control an LO? The hacker would become the "real" owner of the PO and it could control it at will unless the entanglement is broken or unidirectional. In certain situations, like self-driving cars, the communication capabilities and, consequently, the entanglement, are an essential part of the PO, and they could not work or be otherwise usable if the entanglement is not established and operational.

Another important point to evaluate is the applicability of the concept. It is very attractive, but it still needs to demonstrate its merits beyond clear and obvious scenarios, for example, the DT of an aircraft engine is surely a valuable scenario for a manufacturer or a pilot in training, but is it practical or useful for controlling all the airplanes that are operating every day? What about scenarios that have more than one object, or that involve a small set of objects? Can the DT scale up to thousands or even more objects? What about scenarios that have more than one actor and stakeholder involved? Is the assumption of having a single DT for all the possible applications a viable one, or is it better to have specialized DTs within different application scenarios? For instance, the virtual patient option is a general one; should it be specialized for a single major or a small set of diseases? To what extent could the composability be exerted? For instance, should a virtual patient be considered within a smart building application, or would that be too complicated or useless for the goals and means of the smart building?

Many more application scenarios will have to be considered and demonstrated in order to move toward a generalized global architecture of the DT or to a set of specialized applications and platforms for specific problem domains. Experiments, demonstrations, and deployments are needed in order to fully understand the viability, the real possibility to implement, the complexity, and the value of the concept.

From a business perspective, the DT approach appears to be a very useful enabler for the digital transformation/softwarization of several industries. The servitization capability certainly is appealing, useful, and clear enough to be a successful mechanism for supporting new businesses. But, an obvious and very important question must be addressed: what is the user acceptance of this approach? Are customers ready to adopt it? For instance, is a real patient willing to be continuously scrutinized in terms of behavior, actions and status? Is the patient willing to be transformed into a virtual patient? What is the actual value and acceptability of the DT from the customers' perspectives?

As for other pervasive technologies, there is a "huge" privacy issue for the DT: the Servitization of generally used goods and products will inevitably lead to knowledge about people's usage patterns. Specific people's behavior will be revealed to applications and to owners of those applications in a deeper way than what is already possible. In addition, if the DT paradigm is largely applied to several different application domains, a sort of "Virtual User," that is, a counterpart of the physical user, could be created, represented, updated and exploited. It will be used to understand the behavioral patterns of people, and possibly to influence, control, govern or manipulate their behavior. These and other issues and ethical questions must be understood. Regulation mechanisms have to be established beforehand in order to exploit the benefits of this technology while limiting its drawbacks and preventing its misuse (see [207], [289], and [290]).

In order to anticipate the possibilities offered by the DT concept and to better understand the possible evolution of the concept in a medium longer-term perspective, the "Future Characterization" approach proposed in [291] is used here. Four different future options for the DT are considered and briefly discussed.

1) The probable future, that is, the projected baseline. This option identifies the "business as usual" possibilities associated with the DT concept. Under this perspective, some products, from design to

prototyping, from production to operation, will be created, developed, and monitored according to the DT concept. They may be complex objects (an aircraft engine, a car, etc.). The DT is, however, confined to very complex manufacturing format. It is very specialized per kind of (complex) object and it requires a strict control and monitoring during the life cycle. Vertical markets supported by specialized platforms are the target application domains for this approach.

2) The plausible future, that is, what could actually happen. More complex aggregations of components and objects are considered for this option. The scenario of the Industrial Internet or Industry 4.0 seems to be a very likely opportunity for the wide application of the DT concept. Its use will progressively become a best practice in order to manufacture and control products during their entire life cycle (including customer usage). Interoperability and standardization will occur as part of the evolution to ease the interworking between different production systems and application domains. The servitization capabilities will mainly be used for monitoring or for enabling the basic capabilities of the final products. Servitization will prove valid for an increasing number of business offerings and propositions.

3) The possible future, that is, what might happen? The smart city scenario represents this option: different POs need to interact; different and complex aggregations of objects need to be designed, controlled, and managed by utilizing the DT concept. Scaling up in terms of the number of objects to be controlled must be enabled. Interfaces and functions spanning more than one specific application domain need to be created. A large number of interaction issues will need to be addressed by tools and systems. A rich set of standard specifications and interfaces will need to be developed and supported by an open-source software platform capable of allowing different levels of programmability for a large number of objects, components, and functions. In addition, a full ecosystem of actors needs to be involved in the evolution of the DT concept in the smart city context, and a wide range of applications will be delivered to final customers. Servitization will be a major trend and asset for users.

4) The preferable future, that is, what should happen? In this case, the reference scenario is the Programmable World [292], [293], that is, an environment in which each PO is offering APIs by means of its software counterpart. In other words, a large part of the physical world can be represented by means of softwarized objects that can monitor their physical counterpart and be aggregated by different players in order to create applications, or to be used in specialized systems to provide services to customers. This environment could be a sort of "Software Nirvana" where the softwarization has finally taken over the processes of many industries, and the majority of products are associated or even substituted by services. Servitization, then, will be the norm and new business models as well as new ways to conceive products, services, their use, and monetizing will change the shape of traditional business relationships. This scenario sketches out a world that may be profoundly different from todays.

It is difficult to predict the future of the DT concept; however, it is a concept that is already having a significant impact on the manufacturing industry and in IoT, and it has the possibility to have a very large impact on the lives of many people in the future. ■

REFERENCES

[1] M. Grieves, *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*. New York, NY, USA: McGraw-Hill, 2006.

[2] S. Haag and R. Anderl, "Digital twin—Proof of concept," *Manuf. Lett.*, vol. 15, pp. 64–66, Jan. 2018.

[3] R. A. A. M. Biru and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," in *Proc. IEEE Internet Things Initiative*, Piscataway, NJ, USA, May 2015, pp. 1–86. [Online]. Available: https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

[4] D. Riemer, "Feeding the digital twin: Basics, models and lessons learned from building an IoT analytics toolbox (Invited Talk)," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, p. 4212.

[5] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," Univ. Florida, Melbourne, FL, USA, White Paper 1411, 2014, pp. 1–7.

[6] M. Grieves and J. Vickers, "Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*. Berlin, Germany: Springer, 2017 pp. 85–113. [Online]. Available: https://www.springerprofessional.de/en/digital-twin-mitigating-unpredictable-undesirable-emergent-behav/10588328

[7] R. Casadei and M. Viroli, "Collective abstractions and platforms for large-scale self-adaptive IoT," in *Proc. IEEE 3rd Int. Workshops Found. Appl. Self Syst. (FAS*W)*, Sep. 2018, pp. 106–111.

[8] I. Bedhief, L. Foschini, P. Bellavista, M. Kassar, and T. Aguili, "Toward self-adaptive software defined fog networking architecture for IIoT and industry 4.0," in *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–5.

[9] B. A. Talkhestani, N. Jazdi, W. Schloegl, and M. Weyrich, "Consistency check to synchronize the digital twin of manufacturing automation based on anchor points," *Procedia CIRP*, vol. 72, pp. 159–164, Jan. 2018, doi: 10.1016/j.procir.2018.03.166.

[10] H. Zipper and C. Diedrich, "Synchronization of industrial plant and digital twin," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 1678–1681.

[11] Y. Gao, H. Lv, Y. Hou, J. Liu, and W. Xu, "Real-time modeling and simulation method of digital twin production line," in *Proc. IEEE 8th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, May 2019, pp. 1639–1642.

[12] U. Dahmen and J. Rossmann, "Experimentable digital twins for a modeling and simulation-based engineering approach," in *Proc. IEEE Int. Syst. Eng. Symp. (ISSE)*, Oct. 2018, pp. 1–8.

[13] M. E. Porter and J. E. Heppelmann, "How smart, connected products are transforming companies," *Harvard Bus. Rev.*, vol. 93, no. 10, pp. 96–114, 2015.

[14] A. Rotem-Gal-Oz. (2006). *Fallacies of Distributed Computing Explained*. White Paper. Accessed: Feb. 18, 2020. [Online]. Available: http://www.rgoarchitects.com/Files/fallacies.pdf

[15] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[16] M. Kunath and H. Winkler, "Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process," *Procedia CIRP*, vol. 72, pp. 225–231, Jan. 2018.

[17] J. Bao, D. Guo, J. Li, and J. Zhang, "The modelling and operations for the digital twin in the context of manufacturing," *Enterprise Inf. Syst.*, vol. 13, no. 4, pp. 534–556, Apr. 2019.

[18] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Ann.*, vol. 66, no. 1, pp. 141–144, 2017.

[19] E. M. Kraft, "The air force digital thread/digital

twin–life cycle integration and use of computational and experimental knowledge," in *Proc. 54th AIAA Aerosp. Sci. Meeting*, Jan. 2016, p. 0897.
[20] Q. Qi, D. Zhao, T. W. Liao, and F. Tao, "Modeling of cyber-physical systems and digital twin based on edge computing, fog computing and cloud computing towards smart manufacturing," in *Proc. Additive Manuf., Bio Sustain. Manuf.*, vol. 1, Jun. 2018, pp. 1–7.
[21] S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in *Mechatronic Futures*. Berlin, Germany: Springer, 2016, pp. 59–74.
[22] G. Marcus, F. Rossi, and M. Veloso, "Beyond the turing test," *AI Mag.*, vol. 37, no. 1, pp. 3–4, 2016.
[23] J. Scheibmeir and Y. Malaiya, "An API development model for digital twins," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 518–519, doi: 10.1109/QRS-C.2019.00103.
[24] A. Burghardt, D. Szybicki, P. Gierlak, K. Kurc, P. Pietruś, and R. Cygan, "Programming of industrial robots using virtual reality and digital twins," *Appl. Sci.*, vol. 10, no. 2, p. 486, Jan. 2020.
[25] S. Malakuti and S. Grüner, "Architectural aspects of digital twins in IIoT systems," in *Proc. 12th Eur. Conf. Softw. Archit. Companion (ECSA)*, 2018, pp. 1–2.
[26] V. Souza, R. Cruz, W. Silva, S. Lins, and V. Lucena, "A digital twin architecture based on the industrial Internet of Things technologies," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–2, doi: 10.1109/ICCE.2019.8662081.
[27] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
[28] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.
[29] T. Borangiu, D. Trentesaux, A. Thomas, P. Leitão, and J. Barata, "Digital transformation of manufacturing through cloud services and resource virtualization," *Comput. Ind.*, vol. 108, pp. 150–162, Jun. 2019, doi: 10.1016/j.compind.2019.01.006.
[30] M. Ciavotta, M. Alge, S. Menato, D. Rovere, and P. Pedrazzoli, "A microservice-based middleware for the digital factory," *Procedia Manuf.*, vol. 11, pp. 931–938, Jan. 2017.
[31] S. Yun, J.-H. Park, and W.-T. Kim, "Data-centric middleware based digital twin platform for dependable cyber-physical systems," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2017, pp. 922–926.
[32] D. Schela *et al.*, "Manufacturing service bus: An implementation," in *Proc. 11th CIRP Conf. Intell. Comput. Manuf. Eng.*, Jul. 2017, pp. 179–184.
[33] M. Schluse and J. Rossmann, "From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems," in *Proc. IEEE Int. Symp. Syst. Eng. (ISSE)*, Oct. 2016, pp. 1–6, doi: 10.1109/SysEng.2016.7753162.
[34] G. Schroeder *et al.*, "Visualising the digital twin using Web services and augmented reality," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2016, pp. 522–527.
[35] M. Kritzler, M. Funk, F. Michahelles, and W. Rohde, "The virtual twin: Controlling smart factories using a spatially-correct augmented reality representation," in *Proc. 7th Int. Conf. Internet Things (IoT)*, ACM, New York, NY, USA, 2017, pp. 1–2, doi: 10.1145/3131542.3140274.
[36] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," *Procedia Manuf.*, vol. 11, pp. 939–948, Jan. 2017.
[37] B. A. Talkhestani *et al.*, "An architecture of an intelligent digital twin in a cyber-physical production system," *Automatisierungstechnik*, vol. 67, no. 9, pp. 762–782, 2019, doi: 10.1515/auto-2019-0039.
[38] G. Holmes, "Data trends: My digital twin," *Preview*, no. 186, p. 39, Feb. 2017. [Online]. Available: http://www.publish.csiro.au/PV/pdf/PVv2017n186p39
[39] S. Hippold, (Jan. 2019). *How Digital Twins Simplify the IoT*. Gartner, U.K. [Online]. Available: https://www.gartner.com/smarterwithgartner/how-digital-twins-simplify-the-iot/
[40] C. Steinmetz, A. Rettberg, F. G. C. Ribeiro, G. Schroeder, and C. E. Pereira, "Internet of Things ontology for digital twin in cyber physical systems," in *Proc. 8th Brazilian Symp. Comput. Syst. Eng. (SBESC)*, Nov. 2018, pp. 154–159, doi: 10.1109/SBESC.2018.00030.
[41] J. Sleuters, Y. Li, J. Verriet, M. Velikova, and R. Doornbos, "A digital twin method for automated behavior analysis of large-scale distributed IoT systems," in *Proc. 14th Annu. Conf. Syst. Syst. Eng. (SoSE)*, May 2019, pp. 7–12, doi: 10.1109/SYSOSE.2019.8753845.
[42] E. Y. Song, M. Burns, A. Pandey, and T. Roth, "IEEE 1451 smart sensor digital twin federation for IoT/CPS research," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2019, pp. 1–6, doi: 10.1109/SAS.2019.8706111.
[43] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *Proc. 4th ACM Workshop Cyber-Physical Syst. Secur. CPSS*, 2018, pp. 61–72.
[44] C. Maiwald, "With Spime to circular service design: Introducing service design to an IoT platform provider's delivery process," M.S. thesis, Service Innov. Design, Laurea Univ., Uusimaa, Finland, May 2018.
[45] C. Friedow, M. Völker, and M. Hewelt, "Integrating IoT devices into business processes," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Berlin, Germany: Springer, 2018, pp. 265–277.
[46] A. Canedo, "Industrial IoT lifecycle via digital twins," in *Proc. 11th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES)*, 2016, p. 1.
[47] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia CIRP*, vol. 61, pp. 335–340, Jan. 2017.
[48] R. Bohlin, J. Hagmar, K. Bengtsson, L. Lindkvist, J. S. Carlson, and R. Söderberg, "Data flow and communication framework supporting digital twin for geometry assurance," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, 2017, Art. no. V002T02A110.
[49] Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," *AIP Conf. Proc.*, vol. 1949, no. 1, 2018, Art. no. 020023.
[50] S. O. Erikstad, "Merging physics, big data analytics and simulation for the next-generation digital twins," in *Proc. 11th Symp. High-Perform. Mar. Vehicles (HIPER)*, Zevenwacht, South Africa, Sep. 2017, pp. 140–150. [Online]. Available: http://data.hiper-conf.info/Hiper2017_Zevenwacht.pdf
[51] E. Kharlamov, F. Martin-Recuerda, B. Perry, D. Cameron, R. Fjellheim, and A. Waaler, "Towards semantically enhanced digital twins," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 4189–4193, doi: 10.1109/BigData.2018.8622503.
[52] J. Lieberman, A. Leidner, G. Percivall, and C. Ronsdorf, "Using big data analytics and IoT principles to keep an eye on underground infrastructure," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 4592–4601, doi: 10.1109/BigData.2017.8258503.
[53] D. Schmalstieg, and T. Hollerer, *Augmented Reality: Principles and Practice*. 1st ed. Boston, MA, USA: Addison-Wesley, 2016.
[54] A. El Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE Multimedia Mag.*, vol. 25, no. 2, pp. 87–92, Apr. 2018.
[55] K. Börner, "Twin worlds: Augmenting, evaluating, and studying three-dimensional digital cities and their evolving communities," in *Digital Cities II: Computational and Sociological Approaches* (Lecture Notes in Computer Science), vol. 2362. Berlin, Germany: Springer, 2001, pp. 257–269.
[56] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "The 5G-enabled tactile Internet: Applications, requirements, and architecture," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.
[57] T. Boellstorff, "Virtuality. Placing the virtual body: Avatar, chora, cypherg," in *A Companion to Anthropology Body Embodiment*, vol. 29. Hoboken, NJ, USA: Wiley, 2011, pp. 504–520.
[58] M. S. Meadows, *I, Avatar: The Culture and Consequences of Having a Second Life*. Berkeley, CA, USA, New Riders, 2007.
[59] T. M. Malaby, *Making Virtual Worlds: Linden Lab and Second Life*. Ithaca, NY, USA: Cornell Univ. Press, 2009.
[60] J. Wiecha, R. Heyden, E. Sternthal, and M. Merialdi, "Learning in a virtual world: Experience with using second life for medical education," *J. Med. Internet Res.*, vol. 12, no. 1, p. e1, Jan. 2010, doi: 10.2196/jmir.1337.
[61] X. Wang, M. J. Kim, P. Love, and S.-C. Kang, "Augmented reality in built environment: Classification and implications for future research," *Autom. Construct.*, vol. 32, pp. 1–13, Jul. 2013, doi: 10.1016/j.autcon.2012.11.021.
[62] T. Hilfert and M. König, *Low-Cost Virtual Reality Environment for Engineering and Construction*, no. 1. Berlin, Germany: Springer, 2016. [Online]. Available: https://viejournal.springeropen.com/track/pdf/10.1186/s40327-015-0031-5, doi: 10.1186/s40327-015-0031-5.
[63] V. Y. Kharitonov, "A software architecture for high-level development of component-based distributed virtual reality systems," in *Proc. IEEE 37th Annu. Comput. Softw. Appl. Conf.*, Jul. 2013, pp. 696–705, doi: 10.1109/COMPSAC.2013.111.
[64] G. Reitmayr and D. Schmalstieg, "An open software architecture for virtual reality interaction," in *Proc. ACM Symp. Virtual Reality Softw. Technol. (VRST)*, 2001, pp. 47–54, doi: 10.1145/505008.505018.
[65] M. Gerla, D. Maggiorini, C. E. Palazzi, and A. Bujari, "A survey on interactive games over mobile networks," *Wireless Commun. Mobile Comput.*, vol. 13, no. 3, pp. 212–229, Feb. 2013.
[66] B. H. Thomas, "A survey of visual, mixed, and augmented reality gaming," *Comput. Entertainment*, vol. 10, no. 3, pp. 1–33, Nov. 2012.
[67] W. Cai *et al.*, "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, Jun. 2016.
[68] Y. Litvinova, S. V. Rehm, L. C. Goel, and I. I. Junglas, "Collaborating in Virtual Reality by using Digital Twins," in *Proc. ISPIM Innov. Symp.*, Stockholm, Sweden, Jun. 2018, pp. 1–10.
[69] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira, "VR Juggler: A virtual platform for virtual reality application development," *Proc. IEEE Virtual Reality*, Yokohama, Japan, Mar. 2001, pp. 89–96.
[70] J. Autiosalo, "Platform for industrial Internet and digital twin focused education, research, and innovation: Ilmatar the overhead crane," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 241–244.
[71] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325–20336, 2019.
[72] K. Kravari and N. Bassiliades, "A survey of agent platforms," *J. Artif. Societies Social Simul.*, vol. 18, no. 1, p. 11, 2015.
[73] G. Lombardo, P. Fornacciari, M. Mordonini, M. Tomaiuolo, and A. Poggi, "A multi-agent architecture for data analysis," *Future Internet*, vol. 11, no. 2, p. 49, Feb. 2019.
[74] G. Dorrer and S. Yarovoy, "Use of agent-based modeling for wildfire situations simulation," in *Proc. 3rd Russian–Pacific Conf. Comput. Technol. Appl. (RPC)*, Aug. 2018, pp. 1–4, doi: 10.1109/RPC.2018.8481677.
[75] F. Tao, M. Zhang, Y. Liu, and A. Y. C. Nee, "Digital twin driven prognostics and health management

for complex equipment," *CIRP Ann.*, vol. 67, no. 1, pp. 169–172, 2018.
[76] A.-V. Sitar-Taut *et al.*, "Smart homes for older people involved in rehabilitation activities–reality or dream, acceptance or rejection?" *Balneo Res. J.*, vol. 9, no. 3, pp. 291–298, Sep. 2018.
[77] M. Alam, M. M. Tehranipoor, and U. Guin, "TSensors vision, infrastructure and security challenges in trillion sensor era," *J. Hardw. Syst. Secur.*, vol. 1, no. 4, pp. 311–327, Dec. 2017.
[78] G. Fortino, "Agents meet the IoT: Toward ecosystems of networked smart objects," *IEEE Syst., Man, Cybern. Mag.*, vol. 2, no. 2, pp. 43–47, Apr. 2016.
[79] K. Bakliwal, M. H. Dhada, A. S. Palau, A. K. Parlikad, and B. K. Lad, "A multi agent system architecture to implement collaborative learning for social industrial assets," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1237–1242, 2018.
[80] A. Madni, C. Madni, and S. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, Jan. 2019.
[81] P. Skrzypczyński, "Machine learning for embodied agents: From signals to symbols and actions," in *Proc. Tutorial Signal Process., Algorithms, Archit., Arrangements, Appl. (SPA)*, Poznan, Poland, 2019, pp. 13–15.
[82] D. de Groot and F. Brazier, "Identity management in agent systems," in *Proc. 1st Int. Workshop Privacy Secur. Agent-Based Collaborative Environ. (PSACE)*, 2006, pp. 23–34.
[83] N. Jafari Navimipour and F. Sharifi Milani, "A comprehensive study of the resource discovery techniques in peer-to-peer networks," *Peer-to-Peer Netw. Appl.*, vol. 8, no. 3, pp. 474–492, May 2015.
[84] B. Veloso, B. Malheiro, and J. C. Burguillo, "CloudAnchor: Agent-based brokerage of federated cloud resources," in *Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst.* Cham, Switzerland: Springer, Jun. 2016, pp. 207–218.
[85] A. Vereshchaka and W. Dong, "Dynamic resource allocation during natural disasters using multi-agent environment," in *Proc. Int. Conf. Social Comput., Behav.-Cultural Modeling Predict. Behav. Represent. Modeling Simulation*. Cham, Switzerland: Springer, Jul. 2019, pp. 123–132.
[86] H. Hayashi, "Integrating decentralized coordination and reactivity in MAS for repair-task allocations," in *Proc. Int. Conf. Practical Appl. Agents Multi-Agent Syst.* Cham, Switzerland: Springer, Jun. 2017, pp. 95–106.
[87] Y. Fan, L. Liu, G. Feng, and Y. Wang, "Self-triggered consensus for multi-agent systems with zeno-free triggers," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2779–2784, Oct. 2015.
[88] F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *J. Parallel Distrib. Comput.*, vol. 75, pp. 184–197, Jan. 2015, doi: 10.1016/j.jpdc.2014.08.004.
[89] E. C. Ferrer, "The blockchain: A new framework for robotic swarm systems," in *Proc. Future Technol. Conf.* Cham, Switzerland: Springer, 2018, pp. 1037–1058.
[90] A. Grignard, P. Taillandier, B. Gaudou, D. Vo, N. Huynh, and A. Drogoul, "GAMA 1.6: Advancing the art of complex agent-based modeling and simulation," in *Proc. Int. Conf. Princ. Pract. Multi-Agent Syst.* Berlin, Germany: Springer, 2013, pp. 117–131.
[91] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE-A FIPA-compliant agent framework," *Proc. PAAM*, vol. 99, nos. 97–108, p. 33, 1999.
[92] O. Boissier, R. H. Bordini, J. F. Hubner, A. Ricci, and A. Santi, "Multi-agent oriented programming with JaCaMo," *Sci. Comput. Program.*, vol. 78, no. 6, pp. 747–761, Jun. 2013.
[93] S. Zheng, Q. Zhang, R. Zheng, B.-Q. Huang, Y.-L. Song, and X.-C. Chen, "Combining a multi-agent system and communication middleware for smart home control: A universal control platform architecture," *Sensors*, vol. 17, no. 9, pp. 2135–2158, Sep. 2017.
[94] K. Lingaraj, R. V. Biradar, and V. C. Patil, "Eagilla: An enhanced mobile agent middleware for wireless sensor networks," *Alexandria Eng. J.*, vol. 57, no. 3, pp. 1197–1204, Sep. 2018.
[95] J. Achara, M. Mohiuddin, W. Saab, R. Rudnik, and J.-Y. Le Boudec, "T-RECS: A software testbed for multi-agent real-time control of electric grids," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–4, doi: 10.1109/ETFA.2017.8247706.
[96] M. A. B. Brasil, B. Bosch, F. R. Wagner, and E. P. de Freitas, "Performance comparison of multi-agent middleware platforms for wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 7, pp. 3039–3049, Apr. 2018.
[97] F. Zambonelli *et al.*, "Developing pervasive multi-agent systems with nature-inspired coordination," *Pervasive Mobile Comput.*, vol. 17, pp. 236–252, Feb. 2015.
[98] T. Suganuma, T. Oide, S. Kitagami, K. Sugawara, and N. Shiratori, "Multiagent-based flexible edge computing architecture for IoT," *IEEE Netw.*, vol. 32, no. 1, pp. 16–23, Jan. 2018.
[99] T. Jung, P. Shah, and M. Weyrich, "Dynamic co-simulation of Internet-of-Things-components using a multi-agent-system," *Procedia CIRP*, vol. 72, pp. 874–879, Jan. 2018.
[100] A. M. Uhrmacher and D. Weyns, *Multi-Agent Systems Simulation and Applications*. New York, NY, USA: Taylor & Francis, 2009.
[101] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2005.
[102] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, no. 5, pp. 862–876, Apr. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128609003387
[103] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
[104] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.
[105] P. Heidari, Y. Lemieux, and A. Shami, "QoS assurance with light virtualization—A survey," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 558–563, doi: 10.1109/CloudCom.2016.0097.
[106] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, and A. Campi, "Clouds of virtual machines in edge networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 63–70, Jul. 2013.
[107] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 553–576, 1st Quart., 2016.
[108] C. Pahl, P. Jamshidi, and O. Zimmermann, "Architectural principles for cloud software," *ACM Trans. Internet Technol.*, vol. 18, no. 2, pp. 1–23, Mar. 2018.
[109] ETSI. (2013). *Network Function Virtualization: Architectural Framework*. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf
[110] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov. 2014.
[111] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.
[112] S. Vural *et al.*, "Performance measurements of network service deployment on a federated and orchestrated virtualisation platform for 5G experimentation," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2018, pp. 1–6, doi: 10.1109/NFV-SDN.2018.8725755.
[113] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.
[114] H. Ishii and B. Ullmer, "Tangible bits: Towards seamless interfaces between people, bits and atoms," in *Proc. ACM CHI*, vol. 97, 1997, pp. 234–241.
[115] T. DebRoy, W. Zhang, J. Turner, and S. S. Babu, "Building digital twins of 3D printing machines," *Scripta Mater.*, vol. 135, pp. 119–124, Jul. 2017.
[116] R. Minerva, "From Internet of Things to the virtual continuum: An architectural view," in *Proc. Eur Med Telco Conf. (EMTC)*, Nov. 2014, pp. 1–6, doi: 10.1109/EMTC.2014.6996633.
[117] S. Bosse and U. Engel, "Real-time human-in-the-loop simulation with mobile agents, chat bots, and crowd sensing for smart cities," *Sensors*, vol. 19, no. 20, p. 4356, Oct. 2019.
[118] M. Weyrich and C. Ebert, "Reference architectures for the Internet of Things," *IEEE Softw.*, vol. 33, no. 1, pp. 112–116, Jan. 2016.
[119] E. Escorsa, *Digital Twins: A Glimpse at the Main Patented Developments*. Barcelona, Spain: IALE Tecnologia, Aug. 2018. [Online]. Available: https://ificlaims.com/uploads/filedump/partners/IALE_IFIClaims_Post_Digital_Twins_Patents.pdf?1537290452
[120] A. Bolton *et al.*, *Gemini Principles*. Cambridge, U.K.: Univ. Cambridge, 2018. [Online]. Available: https://www.repository.cam.ac.uk/bitstream/handle/1810/284889/The%20Gemini%20Principles%202019.pdf?sequence=7]
[121] A. M. Miller, R. Alvarez, and N. Hartman, "Towards an extended model-based definition for the digital twin," *Comput.-Aided Design Appl.*, vol. 15, no. 6, pp. 880–891, Nov. 2018.
[122] R. A. Freitas and R. C. Merkle, *Kinematic Self-Replicating Machines*. Georgetown, TX, USA: Landes Biosci, 2004.
[123] T. Okita, T. Kawabata, H. Murayama, N. Nishino, and M. Aichi, "A new concept of digital twin of artifact systems: Synthesizing monitoring/inspections, physical/numerical models, and social system models," *Procedia CIRP*, vol. 79, pp. 667–672, Jan. 2019.
[124] S. Boschert, C. Heinrich, and R. Rosen, "Next generation digital Twin," in *Proc. TMCE*, Las Palmas de Gran Canaria, Spain, 2018, pp. 209–218.
[125] J. Van Os, "The digital twin throughout the lifecycle," in *SNAME Maritime Convention*. New York, NY, USA: The Society of Naval Architects and Marine Engineers, Nov. 2018.
[126] *Overview of Ubiquitous Networking and of Its Support in NGN*, document ITU Y.2002, ITU-T Recommendation, 2009.
[127] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1778–1785, doi: 10.1109/CVPR.2009.5206772.
[128] Z. Khalid, N. Fisal, and M. Rozaini, "A survey of middleware for sensor and network virtualization," *Sensors*, vol. 14, no. 12, pp. 24046–24097, Dec. 2014.
[129] M. Eckhart and A. Ekelhart, "A specification-based state replication approach for digital twins," in *Proc. Workshop Cyber-Phys. Syst. Secur. PrivaCy (CPS-SPC)*, 2018, pp. 36–47.
[130] M. Song *et al.*, "*In-situ* AI: Towards autonomous and incremental deep learning for IoT systems," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2018, pp. 92–103, doi: 10.1109/HPCA.2018.00018.
[131] T. H.-J. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper, "The digital twin: Demonstrating the potential of real time data acquisition in production systems," *Procedia Manuf.*, vol. 9, pp. 113–120, Jan. 2017.
[132] J. S. Gilmore and H. A. Engelbrecht, "A survey of

state persistency in peer-to-peer massively multiplayer online games," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 818–834, May 2012.

[133] C. Esposito, A. Castiglione, F. Pop, and K.-K.-R. Choo, "Challenges of connecting edge and cloud computing: A security and forensic perspective," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 13–17, Mar. 2017.

[134] Y. Arfat and F. E. Eassa, "A survey on fault tolerant multi agent system," *IJ Inf. Technol. Comput. Sci.*, vol. 9, pp. 39–48, 2016.

[135] B. Holenstein, B. Highleyman, and P. Holenstein, *Breaking the Availability Barrier II: Achieving Century Uptimes With Active/Active Systems*. Bloomington, IN, USA: AuthorHouse, May 2007.

[136] N. Ivaki, N. Laranjeiro, and F. Araujo, "A survey on reliable distributed communication," *J. Syst. Softw.*, vol. 137, pp. 713–732, Mar. 2018.

[137] J. Grover and R. M. Garimella, "Reliable and fault-tolerant IoT-edge architecture," in *Proc. IEEE Sensors*, Oct. 2018, pp. 1–4.

[138] N. Mohammadi and J. E. Taylor, "Smart city digital twins," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–5, doi: 10.1109/SSCI.2017.8285439.

[139] A. Held, S. Buchholz, and A. Schill, "Modeling of context information for pervasive computing applications," in *Proc. 6th World Multiconf. Systemics, Cybern. Inform.*, 2002, pp. 167–180.

[140] D. Gotz, S. Sun, and N. Cao, "Adaptive contextualization: Combating bias during high-dimensional visualization and data selection," in *Proc. 21st Int. Conf. Intell. User Interfaces IUI*, 2016, pp. 85–95.

[141] A. Siddiqa *et al.*, "A survey of big data management: Taxonomy and state-of-the-art," *J. Netw. Comput. Appl.*, vol. 71, pp. 151–166, Aug. 2016.

[142] R. Barthel, K. Leder Mackley, A. Hudson-Smith, A. Karpovich, M. de Jode, and C. Speed, "An Internet of old things as an augmented memory system," *Pers. Ubiquitous Comput.*, vol. 17, no. 2, pp. 321–333, Feb. 2013.

[143] R. Canetti and M. Fischlin, "Universally composable commitments," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2001, pp. 19–40.

[144] D. Hofheinz and V. Shoup, "GNUC: A new universal composability framework," *J. Cryptol.*, vol. 28, no. 3, pp. 423–508, Jul. 2015.

[145] G. T. Heineman and W. T. Councill, Eds., *Component-Based Software Engineering: Putting the Pieces Together*. Reading, MA, USA: Addison-Wesley, 2001.

[146] R. van Ommering, F. van der Linden, J. Kramer, and J. Magee, "The Koala component model for consumer electronics software," *Computer*, vol. 33, no. 3, pp. 78–85, Mar. 2000.

[147] T. Vale, I. Crnkovic, E. S. de Almeida, P. A. D. M. S. Neto, Y. C. Cavalcanti, and S. R. D. L. Meira, "Twenty-eight years of component-based software engineering," *J. Syst. Softw.*, vol. 111, pp. 128–148, Jan. 2016.

[148] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber–physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 13–28, Jan. 2012.

[149] K. Bakshi, "Microservices-based software architecture and approaches," in *Proc. IEEE Aerosp. Conf.*, Mar. 2017, pp. 1–8, doi: 10.1109/AERO.2017.7943959.

[150] B. Burns and D. Oppenheimer, "Design patterns for container-based distributed systems," in *Proc. 8th USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2016, pp. 1–6. [Online]. Available: https://www.usenix.org/system/files/conference/hotcloud16/hotcloud16_burns.pdf

[151] R. Buyya, M. A. Rodriguez, A. N. Toosi, and J. Park, "Cost-efficient orchestration of containers in clouds: A vision, architectural elements, and future directions," *J. Phys., Conf. Ser.*, vol. 1108, Nov. 2018, Art. no. 012001, doi: 10.1088/1742-6596/1108/1/012001.

[152] A. Tolk, "Interoperability, composability, and their implications for distributed simulation: Towards mathematical foundations of simulation interoperability," in *Proc. IEEE/ACM 17th Int. Symp. Distrib. Simulation Real Time Appl.*, Oct. 2013, pp. 3–9, doi: 10.1109/DS-RT.2013.8.

[153] C. M. Macal, "Everything you need to know about agent-based modelling and simulation," *J. Simul.*, vol. 10, no. 2, pp. 144–156, May 2016.

[154] J. Vom Brocke, M. Petry, and T. Gonser, "Business process management," in *A Handbook of Business Transformation Management Methodology*, vol. 8. Abingdon, U.K.: Routledge, Apr. 2016, pp. 137–172.

[155] E. Silva, T. Batista, and F. Oquendo, "A mission-oriented approach for designing system-of-systems," in *Proc. 10th Syst. Syst. Eng. Conf. (SoSE)*, May 2015, pp. 346–351, doi: 10.1109/SYSOSE.2015.7151951.

[156] E. Bottani, A. Cammardella, T. Murino, and S. A. Vespoli, "From the cyber-physical system to the digital twin: The process development for behaviour modelling of cyber guided vehicle in M2M logic," in *Proc. 12th Summer School Francesco Turco Ind. Syst. Eng.*, 2017, pp. 1–7.

[157] R. Giaffreda, "iCore: A cognitive management framework for the Internet of Things," in *The Future Internet Assembly*. Berlin, Germany: Springer, 2013, pp. 350–352.

[158] A. J. Redelinghuys, K. Kruger, and A. Basson, "A six-layer architecture for digital twins with aggregation," in *Proc. Int. Workshop Service Orientation Holonic Multi-Agent Manuf.* Cham, Switzerland: Springer, Oct. 2019, pp. 171–182.

[159] R. P. Esteves, L. Z. Granville, and R. Boutaba, "On the management of virtual networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 80–88, Jul. 2013.

[160] E. Salvadori, R. Doriguzzi Corin, A. Broglio, and M. Gerola, "Generalizing virtual network topologies in OpenFlow-based networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6, doi: 10.1109/GLOCOM.2011.6134525.

[161] A. Mayoral, R. Vilalta, R. Casellas, R. Martinez, and R. Munoz, "Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances," in *Proc. 42nd Eur. Conf. Opt. Commun. (ECOC)*, Dusseldorf, Germany, Sep. 2016, pp. 1–3.

[162] B. Rusti, H. Stefanescu, M. Iordache, J. Ghenta, C. Brezeanu, and C. Patachia, "Deploying smart city components for 5G network slicing," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 149–154.

[163] B. Egger, E. Gustafsson, C. Jo, and J. Son, "Efficiently restoring virtual machines," *Int. J. Parallel Program.*, vol. 43, no. 3, pp. 421–439, Jun. 2015.

[164] L. Cui, Z. Hao, Y. Peng, and X. Yun, "Piccolo: A fast and efficient rollback system for virtual machine clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 8, pp. 2328–2341, Aug. 2017.

[165] Y. Yamato, Y. Nishizawa, S. Nagao, and K. Sato, "Fast and reliable restoration method of virtual resources on OpenStack," *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 572–583, Apr. 2018.

[166] J. V. Pradilla and C. E. Palau, "Micro virtual machines (MicroVMs) for cloud-assisted cyber-physical systems (CPS)," in *Internet of Things*. San Mateo, CA, USA: Morgan Kaufmann, 2016, pp. 125–142.

[167] D. Zhan *et al.*, "A high-performance virtual machine filesystem monitor in cloud-assisted cognitive IoT," *Future Gener. Comput. Syst.*, vol. 88, pp. 209–219, Nov. 2018.

[168] J. Li, D. Li, Y. Ye, and X. Lu, "Efficient multi-tenant virtual machine allocation in cloud data centers," *Tsinghua Sci. Technol.*, vol. 20, no. 1, pp. 81–89, Feb. 2015.

[169] B.-H. Lim, E. Bugnion, and S. W. Devine, "Mechanism for providing virtual machines for use by multiple users," U.S. Patent 9 323 550, Apr. 26, 2016.

[170] M. Ahmad, J. S. Alowibdi, and M. U. Ilyas, "VIoT: A first step towards a shared, multi-tenant IoT infrastructure architecture," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 308–313.

[171] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou, and F. Zhang, "Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors," *Int. J. Prod. Res.*, vol. 57, no. 20, pp. 6315–6334, Oct. 2019.

[172] C. Liu, P. Jiang, and W. Jiang, "Web-based digital twin modeling and remote control of cyber-physical production systems," *Robot. Comput.-Integr. Manuf.*, vol. 64, Aug. 2020, Art. no. 101956.

[173] J. W. Edwards and D. A. Deets, "Development of a remote digital augmentation system and application to a remotely piloted research vehicle," NASA, Washington, DC, USA, Tech. Rep. TN D-7941, 1975. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19750012221.pdf

[174] J. Glos Williamson, "Digital augmentation of keepsake objects: A place for interaction of memory, story, and self," M.S. thesis, School Archit. Planning, Massachusetts Inst. Technol., Cambridge, MA, USA, Sep. 1997. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/61536/41234861-MIT.pdf?sequence=2

[175] D. Oliver, D. D. Adam, and A. P. Hudson-Smith, "Living with a digital twin: Operational management and engagement using IoT and mixed realities at UCL's here east campus on the queen elizabeth olympic park," in *Proc. 26th Annu. GIScience Res. UK Conf., GISRUK GIS Res. UK (GISRUK)*, Univ. Leicester, Leicester, U.K, 2018, pp. 1–6. [Online]. Available: http://discovery.ucl.ac.uk/10056521/1/Hudson-Smith_Final_Oliver_Dawkins_Digital_Twin_GISRUK_2018%20%281%29_segment.pdf

[176] K. Borodulin, G. Radchenko, A. Shestakov, L. Sokolinsky, A. Tchernykh, and R. Prodan, "Towards digital twins cloud platform," in *Proc. 10th Int. Conf. Utility Cloud Comput.*, Dec. 2017, pp. 209–210.

[177] T. Niemirepo, M. Sihvonen, V. Jordan, and J. Heinila, "Service platform for automated IoT service provisioning," in *Proc. 9th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2015, pp. 325–329.

[178] Ø.J. Rødseth and A. J. Berre, "From digital twin to maritime data space: Transparent ownership and use of ship information," in *Proc. 13th Int. Symp. Integr. Ship's Inf. Syst. Mar. Traffic Eng. Conf. (ISIS-MTE)*, Berlin, Germany, Sep. 2018, pp. 1–8.

[179] J. G. Berti and L. S. Deluca, "Digital agreement management on digital twin ownership change inventors," U.S. Patent 16 392 426, Aug. 15, 2019. Accessed: Feb. 22, 2020. [Online]. Available: https://patentimages.storage.googleapis.com/6c/8a/26/9e9ed9f1d6a043/US20190251575A1.pdf

[180] C. Altun, B. Tavli, and H. Yanikomeroglu, "Liberalization of digital twins of IoT-enabled home appliances via blockchains and absolute ownership rights," *IEEE Commun. Mag.*, vol. 57, no. 12, pp. 65–71, Dec. 2019.

[181] S. Vandermerwe and J. Rada, "Servitization of business: Adding value by adding services," *Eur. Manage. J.*, vol. 6, no. 4, pp. 314–324, Dec. 1988.

[182] C. Kowalkowski, H. Gebauer, B. Kamp, and G. Parry, "Servitization and deservitization: Overview, concepts, and definitions," *Ind. Marketing Manage.*, vol. 60, pp. 4–10, Jan. 2017.

[183] T. S. Baines, H. W. Lightfoot, O. Benedettini, and J. M. Kay, "The servitization of manufacturing: A review of literature and reflection on future challenges," *J. Manuf. Technol. Manage.*, vol. 20, no. 5, pp. 547–567, Jun. 2009.

[184] Q. Qi, F. Tao, Y. Zuo, and D. Zhao, "Digital twin service towards smart manufacturing," *Procedia CIRP*, vol. 72, pp. 237–242, Jan. 2018.

[185] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *Int. J.*

Adv. Manuf. Technol., vol. 94, nos. 9–12, pp. 3563–3576, Feb. 2018.

[186] T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz, "A simulation-based architecture for smart cyber-physical systems," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2016, pp. 374–379, doi: 10.1109/ICAC.2016.29.

[187] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. Air Force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf. 14th AIAA*, Apr. 2012, p. 1818.

[188] T. Jung, N. Jazdi, and M. Weyrich, "A survey on dynamic simulation of automation systems and components in the Internet of Things," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–4.

[189] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, 2015.

[190] M. Castelluccio, "Digital twins invade industry," in *Strategic Finance*. New York, NY, USA: Academic, Mar. 2018, p. 63.

[191] J. David, A. Lobov, and M. Lanz, "Leveraging digital twins for assisted learning of flexible manufacturing systems," in *Proc. IEEE 16th Int. Conf. Ind. Inform. (INDIN)*, Porto, Portugal, Jul. 2018, pp. 529–535, doi: 10.1109/INDIN.2018.8472083.

[192] S. P. A. Datta, "Emergence of digital twins—Is this the march of reason?" *J. Innov. Manage.*, vol. 5, no. 3, pp. 14–33, Nov. 2017.

[193] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[194] M. Zborowski, "Finding meaning, application for the much-discussed 'digital twin,'" *J. Petroleum Technol.*, vol. 70, no. 6, pp. 26–32, Jun. 2018.

[195] T. D. West and M. Blackburn, "Is digital thread/digital twin affordable? A systemic assessment of the cost of DoD's latest manhattan project," *Procedia Comput. Sci.*, vol. 114, pp. 47–56, Jan. 2017.

[196] M. Batty, "Digital twins," *Environ. Planning B, Urban Analytics City Sci.*, vol. 45, no. 5, pp. 817–820, Sep. 2018.

[197] S. A. P. Kumar, R. Madhumathi, P. R. Chelliah, L. Tao, and S. Wang, "A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance," *J. Reliable Intell. Environ.*, vol. 4, no. 4, pp. 199–209, Dec. 2018.

[198] J. Kraft and S. Kuntzagk, "Engine fleet-management: The use of digital twins from a MRO perspective," in *Proc. ASME Turbo Expo Turbomachinery Tech. Conf. Expo.*, Jun. 2017, Art. no. V001T01A007.

[199] B. Brenner and V. Hummel, "Digital twin as enabler for an innovative digital shopfloor management system in the ESB logistics learning factory at Reutlingen–University," *Procedia Manuf.*, vol. 9, pp. 198–205, Jan. 2017.

[200] W. Kuehn, "Digital twins for decision making in complex production and logistic enterprises," *Int. J. Design Nature Ecodynamics*, vol. 13, no. 3, pp. 262–273, Aug. 2018.

[201] F. Tao, M. Zhang, J. Cheng, and Q. Qi, "Digital twin workshop: A new paradigm for future workshop," *Comput. Integr. Manuf. Syst.*, vol. 23, no. 1, pp. 1–9, Jan. 2017.

[202] H. Pargmann, D. Euhausen, and R. Faber, "Intelligent big data processing for wind farm monitoring and analysis based on cloud-technologies and digital twins: A quantitative approach," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2018, pp. 233–237, doi: 10.1109/ICCCBDA.2018.8386518.

[203] G. Miscevic, E. Tijan, D. Zgaljic, and M. Jardas, "Emerging trends in e-logistics," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 1–6.

[204] K. Shubenkova, A. Valiev, V. Shepelev, S. Tsiulin, and K. H. Reinau, "Possibility of digital twins technology for improving efficiency of the branded service system," in *Proc. Global Smart Ind. Conf. (GloSIC)*, Nov. 2018, pp. 1–7, doi: 10.1109/GloSIC.2018.8570075.

[205] J.-T. Amenyo, "Using digital twins and intelligent cognitive agencies to build platforms for automated CxO future of work," 2018, *arXiv:1808.07627*. [Online]. Available: http://arxiv.org/abs/1808.07627

[206] Y. He, J. Guo, and X. Zheng, "From surveillance to digital twin: Challenges and recent advances of signal processing for industrial Internet of Things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 120–129, Sep. 2018.

[207] K. Bruynseels, F. S. De Sio, and J. van den Hoven, "Digital twins in health care: Ethical implications of an emerging engineering paradigm," *Frontiers Genet.*, vol. 9, p. 31, Feb. 2018, doi: 10.3389/fgene.2018.00031.

[208] Y. Liu *et al.*, "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE Access*, vol. 7, pp. 49088–49101, 2019.

[209] A. Karakra, F. Fontanili, E. Lamine, J. Lamothe, and A. Taweel, "Pervasive computing integrated discrete event simulation for a hospital digital twin," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2018, pp. 1–6, doi: 10.1109/AICCSA.2018.8612796.

[210] M. Macchi, I. Roda, E. Negri, and L. Fumagalli, "Exploring the role of digital twin for asset lifecycle management," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 790–795, 2018.

[211] Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using Digital Twin," *AIP Conf. Proc.*, vol. 1949, Apr. 2018, Art. no. 020023.

[212] C. N. Verdouw, J. W. Kruize, J. Wolfert, and G. Chatzikostas, "Digital twins in farm management: Illustrated by cases from FIWARE accelerators SmartAgriFood and fractals," presented at the 11th Int. Eur. Forum (Igls-Forum) (161st EAAE Seminar) Syst. Dyn. Innov. Food Netw., Igls, Austria, Feb. 2017. [Online]. Available: https://edepot.wur.nl/440043

[213] G. S. Hamilton *et al.*, "Detecting opportunities and challenges for australian rural industries: Final report," AgriFutures Australia, Canberra, ACT, Australia, Tech. Rep. 18/009, Feb. 2018. [Online]. Available: https://www.agrifutures.com.au/wp-content/uploads/2018/04/18-009.pdf

[214] S.-K. Jo, D.-H. Park, H. Park, and S.-H. Kim, "Smart livestock farms using digital twin: Feasibility study," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 1461–1463, doi: 10.1109/ICTC.2018.8539516.

[215] C. Brosinsky, D. Westermann, and R. Krebs, "Recent and prospective developments in power system control centers: Adapting the digital twin technology for application in power system control centers," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Jun. 2018, pp. 1–6, doi: 10.1109/ENERGYCON.2018.8398846.

[216] S. Balachandar and R. Chinnaiyan, "Reliable digital twin for connected footballer," in *Proc. Int. Conf. Comput. Netw. Commun. Technol.* in Lecture Notes on Data Engineering and Communications Technologies, vol. 15. Singapore: Springer, 2019.

[217] C. Constantinescu, D. Popescu, O. Todorovic, O. Virlan, and V. Tinca, "Methodology of realizing the digital twin of exoskeleton-centred workspaces," *Acta Technica Napocensis Ser., Appl. Math., Mech., Eng.*, vol. 29, no. 61, p. 3, Sep. 2018.

[218] I. Verner, D. Cuperman, A. Fang, M. Reitman, T. Romm, and G. Balikin, "Robot online learning through digital twin experiments: A weightlifting project," in *Proc. 14th Int. Conf. Remote Eng. Virtual Instrum. REV*. New York, NY, USA: Columbia Univ., Mar. 2017, pp. 307–314.

[219] W. Swartout *et al.*, "Virtual humans for learning," *AI Mag.*, vol. 34, no. 4, pp. 13–30, 2013.

[220] A. Belkin, "Object-oriented world modelling for autonomous systems," in *Proc. Joint Workshop Fraunhofer IOSB Inst. Anthropomatics*, 2010, p. 231.

[221] D. Leidner, C. Borst, and G. Hirzinger, "Things are made for what they are: Solving manipulation tasks by using functional object classes," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2012, pp. 429–435, doi: 10.1109/HUMANOIDS.2012.6651555.

[222] R. Jabangwe, J. Börstler, D. Šmite, and C. Wohlin, "Empirical evidence on the link between object-oriented measures and external quality attributes: A systematic literature review," *Empirical Softw. Eng.*, vol. 20, no. 3, pp. 640–693, Jun. 2015.

[223] C. Gezer and E. Taskin, "An overview of oneM2M standard," in *Proc. 24th Signal Process. Commun. Appl. Conf. (SIU)*, May 2016, pp. 1705–1708.

[224] *SmartM2M: IoT Standards Landscape*, document ETSI TR 103 375 V1.1.1, Alliance for IoT Innovation, AIOTI, Oct. 2016.

[225] Alliance for IoT (AIOTI). (2018). *High Level Architecture, Release 4.0*. [Online]. Available: https://aioti.eu/wp-content/uploads/2018/06/AIOTI-HLA-R4.0.7.1-Final.pdf

[226] E. Y. Song, M. Burns, A. Pandey, and T. Roth, "IEEE 1451 smart sensor digital twin federation for IoT/CPS research," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2019, pp. 1–6.

[227] J. Kiljander *et al.*, "Semantic interoperability architecture for pervasive computing and Internet of Things," *IEEE Access*, vol. 2, pp. 856–873, 2014.

[228] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective," *J. Netw. Comput. Appl.*, vol. 81, pp. 111–124, Mar. 2017.

[229] K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos, "Which IoT protocol? Comparing standardized approaches over a common M2M application," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.

[230] I. Florea, R. Rughinis, L. Ruse, and D. Dragomir, "Survey of standardized protocols for the Internet of Things," in *Proc. 21st Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2017, pp. 190–196.

[231] C. Sharma and N. K. Gondhi, "Communication protocol stack for constrained IoT systems," in *Proc. 3rd Int. Conf. Internet Things: Smart Innov. Usages (IoT-SIU)*, Feb. 2018, pp. 1–6.

[232] K. Lee and S. Seol, "Applying CoAP for real-time device control over public networks," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2018, pp. 1–2.

[233] P. Bellavista, L. Foschini, N. Ghiselli, and A. Reale, "MQTT-based middleware for container support in fog computing environments," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2019, pp. 1–7.

[234] J.-S. Sung, "IoT lighting address scheme and profile API design for interoperability," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 1008–1011.

[235] N. Sahni, J. Bose, and K. Das, "Web APIs for Internet of Things," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2018, pp. 2175–2181.

[236] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "FogFlow: Easy programming of IoT services over cloud and edges for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 696–707, Apr. 2018.

[237] J. An *et al.*, "Toward global IoT-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5753–5765, Jun. 2019.

[238] M. Eckhart, A. Ekelhart, and E. Weippl, "Enhancing cyber situational awareness for cyber-physical systems through digital twins," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 1222–1225.

[239] M. Kenney and J. Zysman, "The rise of the platform economy," *Issues Sci. Technol.*, vol. 32,

no. 3, p. 61, 2016. [Online]. Available: https://issues.org/the-rise-of-the-platform-economy/

[240] M. Ceccagnoli, C. Forman, P. Huang, and D. J. Wu, "Co-creation of value in a platform ecosystem: The case of enterprise software," *MIS Quart., Forthcoming*, vol. 36, no. 1, pp. 263–290, 2012. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1824389

[241] N. Walravens and P. Ballon, "Platform business models for smart cities: From control and value to governance and public value," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 72–79, Jun. 2013.

[242] J. Oh, B. Koh, and S. Raghunathan, "Value appropriation between the platform provider and app developers in mobile platform mediated networks," *J. Inf. Technol.*, vol. 30, no. 3, pp. 245–259, Sep. 2015.

[243] R. N. Bolton *et al.*, "Customer experience challenges: Bringing together digital, physical and social realms," *J. Service Manage.*, vol. 29, no. 5, pp. 776–808, Oct. 2018.

[244] D. Happ, N. Karowski, T. Menzel, V. Handziski, and A. Wolisz, "Meeting IoT platform requirements with open pub/sub solutions," *Ann. Telecommun.*, vol. 72, nos. 1–2, pp. 41–52, Feb. 2017.

[245] S. Gilbert and N. Lynch, "Perspectives on the CAP theorem," *Computer*, vol. 45, no. 2, pp. 30–36, Feb. 2012, doi: 10.1109/MC.2011.389.

[246] E. Brewer, "Spanner, TrueTime and the CAP theorem," Google, Mountain View, CA, USA, Tech. Rep. 45855, 2017. [Online]. Available: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45855.pdf

[247] C. Kong *et al.*, "ACID encountering the CAP theorem: Two bank case studies," in *Proc. 12th Web Inf. Syst. Appl. Conf. (WISA)*, Sep. 2015, pp. 235–240, doi: 10.1109/WISA.2015.63.

[248] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "SMART on FHIR: A standards-based, interoperable apps platform for electronic health records," *J. Amer. Med. Inform. Assoc.*, vol. 23, no. 5, pp. 899–908, Sep. 2016.

[249] J. DeWitt, *Best Practices for Heterogenous Health IoT Integration Into Electronic Health Records*. Eugene, OR, USA: Univ. Oregon. Accessed: Feb. 29, 2020. [Online]. Available: https://scholarsbank.uoregon.edu/xmlui/bitstream/handle/1794/24787/DeWitt_2019.pdf?sequence=1

[250] R. Cook-Deegan and A. L. McGuire, "Moving beyond bermuda: Sharing data to build a medical information commons," *Genome Res.*, vol. 27, no. 6, pp. 897–901, Jun. 2017.

[251] R. Kumar and M. P. Rajasekaran, "An IoT based patient monitoring system using raspberry pi," in *Proc. Int. Conf. Comput. Technol. Intell. Data Eng. (ICCTIDE)*, Jan. 2016, pp. 1–4.

[252] A. A. Kononowicz, N. Zary, S. Edelbring, J. Corral, and I. Hege, "Virtual patients—What are we talking about? A framework to classify the meanings of the term in healthcare education," *BMC Med. Edu.*, vol. 15, no. 1, p. 11, Dec. 2015.

[253] W. Serrano, "Digital systems in smart city and infrastructure: Digital as a service," *Smart Cities*, vol. 1, no. 1, pp. 134–154, 2018.

[254] M. Batty, *Cities and Complexity: Understanding Cities With Cellular Automata, Agent-Based Models, and Fractals*. Boston, MA, USA: MIT Press, Sep. 2007.

[255] "GE digital twin—Analytic engine for the digital power plant," Gen. Electr. Power Digit. Solutions, San Ramon, CA, USA, White Paper, 2016. [Online]. Available: https://www.ge.com/digital/sites/default/files/download_assets/Digital-Twin-for-the-digital-power-plant-.pdf

[256] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A standard-based open source IoT platform: FIWARE," *IEEE Internet Things Mag.*, vol. 2, no. 3, pp. 12–18, Sep. 2019.

[257] A. Alonso *et al.*, "An identity framework for providing access to FIWARE OAuth 2.0-based services according to the eIDAS European regulation," *IEEE Access*, vol. 7, pp. 88435–88449, 2019.

[258] L. Moor, L. Bitter, M. D. Prado, N. Pazos, and N. Ouerhani, "IoT meets distributed AI—Deployment scenarios of Bonseyes AI applications on FIWARE," in *Proc. IEEE 38th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Oct. 2019, pp. 1–2.

[259] C. Steinmetz *et al.*, "Ontology-driven IoT code generation for FIWARE," in *Proc. IEEE 15th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2017, pp. 38–43.

[260] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall, and M. Zhao, "Standards-based worldwide semantic interoperability for IoT," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 40–46, Dec. 2016.

[261] V. Rampérez, J. Soriano, D. Lizcano, and J. A. Lara, "An innovative approach to improve elasticity and performance of message brokers for green smart cities," in *Proc. 4th Int. Conf. Eng. MIS (ICEMIS)*, 2018, pp. 1–5.

[262] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang, "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 3, pp. 1155–1166, Mar. 2019.

[263] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[264] U. Pontes, Ed, *Visit [Industrial Internet of Things—Digital Twins], Fraunhofer Institute of Optronics*. Karlsruhe, Germany: System—Technologies and Image Exploitation IOSB, 2018. [Online]. Available: https://www.iosb.fraunhofer.de/servlet/is/14330/visIT_1-26-03-2018_web.pdf

[265] "PREDIX the industrial IoT application platform," General Electr. Digit., San Ramon, CA, USA, White Paper, 2017. [Online]. Available: https://www.ge.com/digital/sites/default/files/download_assets/Predix-The-Industrial-Internet-Platform-Brief.pdf

[266] F. Luyckx. *Impact of the Digital Twin on the Enterprise Architecture*. Blog entry. Accessed: Jun. 14, 2018. [Online]. Available: https://vnsg.nl/iMIS/VNSGWeb/Blogs/Impact_of_the_digital_twin_on_the_enterprise_architecture.aspx

[267] "PREDIX the industrial IoT application platform," Gen. Electr., San Ramon, CA, USA, White Paper, 2017. [Online]. Available: https://www.ge.com/digital/sites/default/files/download_assets/Predix-The-Industrial-Internet-Platform-Brief.pdf

[268] H. Zhang, Q. Liu, X. Chen, D. Zhang, and J. Leng, "A digital twin-based approach for designing and multi-objective optimization of hollow glass production line," *IEEE Access*, vol. 5, pp. 26901–26911, 2017.

[269] *The Promise of a Digital Twin Strategy. Best Practices for Designers and Manufacturers of Products and Industrial Equipment*, White paper from Microsoft Services, Microsoft, Redmond, WI, USA, 2017. [Online]. Available: https://info.microsoft.com/rs/157-GQE-382/images/Microsoft%27s%20Digital%20Twin%20%27How-To%27%20Whitepaper.pdf

[270] B. Van Hoof, *Announcing Azure Digital Twins: Create Digital Replicas of Spaces and Infrastructure Using Cloud, AI and IoT*. Microsoft Blog Entry. Accessed: Sep. 24, 2018. [Online]. Available: https://blogs.microsoft.com/iot/2018/09/24/announcing-azure-digital-twins-create-digital-replicas-of-spaces-and-infrastructure-using-cloud-ai-and-iot/

[271] D. Edwards, "IBM digital twin: Designing for a connected, software driven world," in *Digital Twin at the Genius of Things Summit in Boston*. Armonk, NY, USA: IBM Watson, Oct. 2017. [Online]. Available: https://www.slideshare.net/IBMIoT/ibm-digital-twin-designing-for-a-connected-software-driven-world

[272] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner, "An Internet of Things platform for real-world and digital objects," *Scalable Comput., Pract. Exper.*, vol. 13, no. 1, pp. 45–57, 2012.

[273] D. Ammermann. *Digital Twin Implementation*. SAP Blog Entry. Accessed: Sep. 9, 2017. [Online]. Available: https://blogs.sap.com/2017/09/09/digital-twin-implementation/

[274] AWS. (2018). *Meet Preston, and Explore Your Digital Twin in Virtual Reality*. (GPSTEC321)—AWS Presentation at Re: Invent. [Online]. Available: https://www.slideshare.net/AmazonWebServices/meet-preston-and-explore-your-digital-twin-in-virtual-reality-gpstec321-aws-reinvent-2018

[275] *DITTO API Definition and Documentation*, API specification from Eclipse for the Digital Twin, Eclipse, Ottawa, ON, Canada. [Online]. Available: https://www.eclipse.org/ditto/intro-overview.html

[276] R. Höttger, M. Ozcelikors, P. Heisig, L. Krawczyk, P. Cuadra, and C. Wolff, "Combining eclipse IoT technologies for a RPI3-rover along with eclipse Kuksa," in *Proc. Softw. Eng. (Workshops)*, 2018, pp. 101–106.

[277] J. Cheng, W. Chen, F. Tao, and C.-L. Lin, "Industrial IoT in 5G environment towards smart manufacturing," *J. Ind. Inf. Integr.*, vol. 10, pp. 10–19, Jun. 2018.

[278] B. Varghese *et al.*, "Realizing edge marketplaces: Challenges and opportunities," *IEEE Cloud Comput.*, vol. 5, no. 6, pp. 9–20, Nov. 2018.

[279] V. Damjanovic-Behrendt, "A digital twin architecture for security, privacy and safety," *ERCIM NEWS*, vol. 115, pp. 25–26, Oct. 2018.

[280] M. Lammers. (2018). *Digital Twin Offers Huge Opportunities for real Estate Life Cycle*. PropTechNL Blog Entry. [Online]. Available: https://www.proptech.nl/blog/digital-twin-life-cycle/

[281] C. Sarkar, S. N. A. U. Nambi, R. V. Prasad, and A. Rahim, "A scalable distributed architecture towards unifying IoT applications," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 508–513, doi: 10.1109/WF-IoT.2014.6803220.

[282] M. Kiran, P. Murphy, I. Monga, J. Dugan, and S. S. Baveja, "Lambda architecture for cost-effective batch and speed big data processing," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 2785–2792, doi: 10.1109/BigData.2015.7364082.

[283] M. Gribaudo, M. Iacono, and M. Kiran, "A performance modeling framework for lambda architecture based applications," *Future Gener. Comput. Syst.*, vol. 86, pp. 1032–1041, Sep. 2018.

[284] A. Gallidabino *et al.*, "On the architecture of liquid software: Technology alternatives and design space," in *Proc. 13th Work. IEEE/IFIP Conf. Softw. Archit. (WICSA)*, Apr. 2016, pp. 122–127, doi: 10.1109/WICSA.2016.14.

[285] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping nfv-based multi-access edge computing in 5G ready networks with open baton," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2017, pp. 1–4, doi: 10.1109/NETSOFT.2017.8004237.

[286] J. Asenjo and C. Sastry, "Digital twin—Modeling interrelated devices," in *Journal of Innovation*. Needham, MA, USA: Internet Industrial Consortium, Mar. 2018. [Online]. Available: https://www.iiconsortium.org/news/joi-articles/2018-March_Digital_Twin_TCS.pdf

[287] M. Bauer and W. Li, "Standards enabled digital twin in LSP," presented at the ETSI IoT Week, Sophia Antipolis, France: EU project AUTOPILOT, Oct. 2018. [Online]. Available: https://docbox.etsi.org/Workshop/2018/201810_IoTWEEK/02_IoTWORKSHOP/S06_PART2_IoT_IN_ACTION/AUTOPILOT_NEC-BAUER_EGM-LI.pdf

[288] *Standard Progress on IoT Requirements and Framework and Smart Sustainable Cities' Interoperability Report about Question*, Standard ITU-T SG20, 2018. [Online]. Available: https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/Documents/Events/2018/ssceg2018/Presentation%20and%20Bio/Session5_ITU-T%20SG20%20standard%20progress%

20on%20IoT%20requirements%20framework%20-Xueqin.pdf

[289] T. van Zütphen, "Digital twins when decency meets temptation," T-Syst., Frankfurt, Germany, Mar. 2018. [Online]. Available: https://www.t-systems.com/blob/840400/aaf49862e18052649ab56e66c3420896/DL_Best-Practice_03-2018_FT_Topstory.pdf

[290] C. Pettey, (Sep. 18, 2017). *Prepare for the Impact of Digital Twins*. Gartner Trends. [Online]. Available: https://www.gartner.com/smarterwithgartner/prepare-for-the-impact-of-digital-twins/

[291] J. Voros, "Big history and anticipation: Using big history as a framework for global foresight," in *Handbook of Anticipation: Theoretical and Applied Aspects of the Use of Future in Decision Making*, R. Poli, Ed. Cham, Switzerland: Springer, Jan. 2017, doi: 10.1007/978-3-319-31737-3_95-1.

[292] B. Wasik, *In the Programmable World, All Our Objects Will Act as One*. San Francisco, CA, USA: Wired, May 2013. [Online]. Available: http://www.wired.com/2013/05/internet-of-things-2/

[293] A. Taivalsaari and T. Mikkonen, "A roadmap to the programmable world: Software challenges in the IoT era," *IEEE Softw.*, vol. 34, no. 1, pp. 72–80, Jan. 2017.

ABOUT THE AUTHORS

Roberto Minerva (Senior Member, IEEE) received the M.S. degree (*summa cum laude*) in computer science from the University of Bari Aldo Moro, Bari, Italy, in 1987, and the Ph.D. degree in computer science and telecommunications from Pierre and Marie Curie University–Sorbonne University, Paris, France, in 2013.

From 1987 to 1996, he was a Researcher in the area of service architectures and network intelligence with the Telecom Italia Research Center, Turin, Italy. In the following years, he was responsible for several research groups related to network intelligence and evolution to next generation networks. From 2013 to 2016, he was appointed to the Strategic Initiatives of TIM. Since 2016, he has been the Technical Project Leader of the SoftFIRE, a European Project devoted to the experimentation of network function virtualization (NFV), software defined networking (SDN), and edge computing. Since 2018, he has been an Associate Professor in softwarization with the Service Architecture Laboratory, Wireless Networks and Multimedia Services Department, Institut Mines Telecom, Telecom Sud Paris, Évry, France, a part of the Institute Polytechnique de Paris, Paris. He has authored the book *Networks and New Services: A Complete Story*. He is also an author of more than 50 papers in journals and international conferences.

Dr. Minerva is a member of the Scientific Committee of the Fondazione Bruno Kessler, Trento, Italy. He has been the Chairperson of the IEEE Internet of Things Initiative from 2014 to 2016.

Gyu Myoung Lee (Senior Member, IEEE) received the B.S. degree from Hongik University, Seoul, South Korea, in 1999, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2000 and 2007, respectively.

Since 2008, he has been working with the Institut Mines-Telecom, Telecom SudParis, Évry, France. Since 2014, he has been a Reader with Liverpool John Moores University (LJMU), Liverpool, U.K. Since 2012, he has been with the KAIST Institute for Information Technology Convergence, Daejeon, as an Adjunct Professor. Since 2012, he has been invited to work with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. He also worked as a Research Professor with KAIST and a Guest Researcher with the National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, in 2007. His research interests include the Internet of Things, future networks, multimedia services, and energy saving technologies, including smart grids.

Dr. Lee has been actively working for standardization in ITU-T, IETF, and oneM2M. In ITU-T, he currently serves as the WP Chair, the Rapporteur of Q16/13 in SG13 (Future Networks) and Q4/20 in SG20 (IoT), and the Chair of FG-Data Processing and Management (DPM).

Noël Crespi (Senior Member, IEEE) received the master's degree from the Universities of Orsay (Paris 11), Orsay, France, and Kent, U.K., in 1991, the Diplome d'Ingénieur from Telecom ParisTech, Paris, France, in 1993, and the Ph.D. degree and Habilitation from UPMC (Paris-Sorbonne University), Paris, in 2006 and 2009, respectively.

Since 1993, he has been working at CLIP, Paris, then at Bouygues Telecom, Paris, and then at Orange Labs, Issy Le Moulineaux, in 1995. He took leading roles in the creation of new services with the successful conception and launch of Orange prepaid service, and in standardization (from rapporteurship of IN standard to coordination of all mobile standards activities for Orange). In 1999, he joined Nortel Networks, Paris, as the Telephony Program Manager, architecting core network products for the Europe, the Middle East, and Africa (EMEA) region. He joined Institut Mines-Telecom, Evry, France, in 2002, where he is currently a Professor and the Program Director, leading the Service Architecture Lab. He coordinates the standardization activities for Institut Mines-Telecom at ITU-T and ETSI. He is also an Adjunct Professor at the Korea Advanced Institute of Science and Technology (KAIST), Daejong, South Korea, an Affiliate Professor at Concordia University, Montréal, QC, Canada, and a Guest Researcher at the University of Goettingen, Germany. He is also the Scientific Director of the French-Korean distributed laboratory ILLUMINE. His current research interests are in data analytics, the Internet of Things, and softwarisation.