IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 5 May 2021; revised 16 September 2021; accepted 24 November 2021.
Date of publication 6 December 2021; date of current version 4 March 2022.

Digital Object Identifier 10.1109/TETC.2021.3131532

Key-Components for Digital Twin Modeling With Granularity: Use Case Car-as-a-Service

CHARLES STEINMETZ, GREYCE N. SCHROEDER, RICARDO N. RODRIGUES, ACHIM RETTBERG, AND CARLOS E. PEREIRA

Charles Steinmetz and Achim Rettberg are with the Hamm-Lippstadt University of Applied Sciences, 59063 Hamm, Germany
Greyce N. Schroeder and Carlos E. Pereira are with the Department of Electrical Engineering,
Federal University of Rio Grande do Sul, Porto Alegre, RS 90040-060, Brazil
Ricardo N. Rodrigues is with the Center for Computacional Sciences, Universidade Federal do Rio Grande, Rio Grande, RS 91501-970, Brazil
CORRESPONDING AUTHOR: CHARLES STEINMETZ (charles.steinmetz@hshl.de)

ABSTRACT Digital technologies are changing the way people interact with the world. The Digital Twin (DT) is one of the key enablers of Industry 4.0. It provides a virtual representation of an observable element of the real world. These elements can be both physical objects such as devices or non-physical such as interactions and processes. Digitalization of the real world enables new business models, transforming traditional products into services, as for instance, the Car-as-a-Service (CaaS). To integrate all components that will be part of systems like CaaS or Smart Cities, it is necessary to have well-defined standards for modeling and defining an architecture especially taking into consideration the granularity level of the system. This paper proposes the main components needed for building DT-based systems with different levels of granularity. These components have been arranged in layers to specify the concerns of each part of the system. A case study has been developed to demonstrate the modeling and the deployment of the Digital Twin, highlighting how this concept can be one of the key enablers for CaaS.

INDEX TERMS Digital twin, architecture, granularity, model, car-as-a-service, industry 4.0

I. INTRODUCTION

Industry 4.0 aims to improve flexibility along with mass customization, quality, and improve productivity since it combines production systems with intelligent processes that will change industry business models [1].

The Digital Twin (DT) is one key enabler of the Industry 4.0 [2]. DT can be understood as a virtual representation of assets from the real world, which enables digitalization by enabling monitoring and control of these elements [3]. Any object that can be observed (observable asset) can be represented by a virtual model.

DT can be used to enable different kind of applications [4] and expose them as services [2]. One example of this is the DT used to enable Car-as-a-Service (CaaS) in Smart Cities, where digitalization is also essential for the well functioning of the application.

However, this kind of application can also bring more complexity to the system, once they may have different granularity levels or coupling dependencies between assets [5]. Additionally, by integrating different assets, there are several users with distinct responsibilities/capabilities. This increases the need for customization of systems.

New concepts and business models, like CaaS, are going to become reality in the next few years, but software specifications and deployment details still need to be addressed, in the direction of creating standards, methodologies, and modeling languages [6]. Especially because real-world applications are composed of observable assets and they can be represented in several ways and perspectives [7], standards for organizing information and integrating these heterogeneous models are required [8].

The ISO 23247 [9] brings a standardization for DT in manufacturing activities which can also be a base for further approaches. However, to increase the adoption of standards like this, also in other domains, extensions and improvements have to be done. Defining key-components to build a DT and supporting granularity are necessary aspects to be studied.

In this context, this paper proposes a definition of key-components for modeling DT with granularity. These components are presented with their relationships and compositions. They are also arranged in 4 layers, defined by the ISO 23247, to show their responsibilities. A use case of a CaaS application has been developed to demonstrate the proposed contributions.

The paper is organized as follows: Section II brings a review of the state of the art in the context of DT; in section III the key-components are presented with two overviews: the first, arranged in layers and the second, showing their relationship; in section IV a use case used to validate the proposed approach is implemented; in section V a discussion is presented and finally, in section VI conclusions are drawn and future research directions are signaled.

II. LITERATURE REVIEW

In this section, a literature review on DT, its application and modeling standards, has been done. Researches in the direction of standardization of DT have been identified and used as the base for this paper.

A. DIGITAL TWIN

The Digital Twin concept was originally presented by Michael Grieves and in 2003 at the University of Michigan [10]. Since then, academia and the industry have their attention to the Digital Twin model.

A DT can be defined as a digital model that reflects the state of a corresponding system twin that allows simulations of the physical device using physical models, sensor data, fleet history, in order to represent the lifecycle of its corresponding twin [11].

For Rosen [12], the Digital Twin can be characterized as a virtual reflection of a physical asset facilitated through data and simulators for real-time forecast, controlling, optimization, monitoring and improved decision making. The Digital Twin, through the Internet of Things, can provide many benefits as services such as simulations, devices monitoring and management since it can connect both the physical and the virtual worlds and it can store data of the whole life-cycle [13].

B. DIGITAL TWIN MODELING AND DEPLOYMENT

Regarding the modeling and deployment of DTs, there is a lack of standards [8]. Nevertheless, there are several proposals for modeling and construction of DTs available in the literature, which are related to this work.

A dynamic modeling approach of DT using ontologies is proposed in Xu [14] where the relationships between equipment are described by a combination of basic, functional and manufacturing process ontologies. The relationship between parameters and the corresponding data stream of the real device is also mapped in the ontology.

Zhang [15] proposed a model of digital twin based on physical perception data. The model is composed of three parts: the plant physical model, digital model and 3D virtual model. The models are combined in real-time to form a basic digital twin model.

In [16] it is proposed a Reconfigurable Digital Twin (RTD). The RDT model is described from five dimensions including geometry, physics, capability, behavior, and rule. This approach is suitable for building DT systems with high fidelity since it is possible to describe the same asset with different perspectives/dimensions.

A reference architecture named "Digital Twin-based Cyber-Physical Production System" (DT-CPPS) is proposed by Ding [17]. This architecture has been developed to be used as the basis for future DT systems that interconnect a physical shop floor and its corresponding cyber shop floor.

A DT concept using micro-sevices has been proposed by Damjanovic-Behrendt and Behrendt [18]. However, it was not specified which methodology has been used for the implementation and deployment of DT. A similar approach has been presented by Rovere [19], where micro-services have also been used for building a DT architecture. In this approach, a middleware has been used to enable communication between the physical and the digital world. Devices connect to the middleware via WebSockets. The digital twin representation is composed of functional and behavioral models.

Schroeder [20] proposes a methodology for modeling data exchange for general digital twin systems, using AutomationML (Automation Markup Language - AML). Other researches like [21], [22] and [23] also propose the use of AutomationML to create a model for the DT.

A Digital Twin reference architecture model for Industry 4.0 is proposed in [2]. It is also proposed the concept of Digital Twin as a Service (DTaaS) on the top of the proposed architecture model. To achieve this, the authors have adopted RAMI, which is a recognized reference architecture model in Industry 4.0 and it provides an agile approach for integrating DT in different levels. The present work shares some similarities with this work, such as the idea of exposing assets as services that can be consumed by other services or users.

In [24] the problem of how to incorporate the digital twin-driven approach into the reconfiguration of the manufacturing system (RMS) is highlighted. Therefore, a digital twin-based approach for rapid reconfiguration of automated manufacturing systems is proposed. An IIoT model based on REST API is proposed for enabling reconfiguration of controls and sensor network without human intervention. A bi-level model of productivity rebalancing and lower-level reconfiguration is presented to find an optimal reconfiguration solution. However, as the prototype is under development, there are still some limitations in handling out-of-ordinary events and errors. Additionally, AI methods are planned to be integrated to improve the reconfiguration engine.

Redelinghuys [25] presents a sixlayer architecture for DT. This architecture integrates the physical asset with its virtual representation, and it allows to add objects from the external world. The architecture is composed by the layers: local data, IoT Gateway, cloud-based databases, and emulations and simulations. A physical component of a manufacturing system has been implemented to validate the proposed architecture.

In [26], Gehrmann discusses how a digital twin replication model and corresponding security architecture can be used to allow data sharing and control of security-critical processes.

Yun [8] propose a Digital Twin Platform that has scalable and flexible services to different applications, named uDiT (universal Digital Twin platform). This platform has functions like communication middleware based on the object management group, data distribution service, middleware and interface for the DT runtime engine, co-simulation functions based on functional mockup interface, and functions for media and protocol conversion. The work tries to create a platform that allows interconnection between digital twins.

In [5] a survey on the state-of-the-art approaches, that have contributed to DT, is given. Regarding DT architectures, the author concludes that generally, they are composed by layers, which is also the case of the present work. The layers located at the bottom of the architectures are the ones that interact with the devices and provide ways to sense and control them. Functions or methods are necessary to allocate resources and for virtualizing the assets in the infrastructure. The upper layers contain mainly the models and data representations of the objects contained in the bottom layers. Data is collected and contextualized to be used for analysis and information inferring. There may exist data that comes from outside of the set of physical objects, such as from external systems. Ontologies can be added to represent a domain-specific semantic. Simulations are on the top of all these layers and APIs provide an interface to the applications.

It was proposed by Souza [27] a DT architecture applying IIoT technologies for sense and actuate on the DT, the use the OPC UA communication protocol for structure and exchange unified data, and provide services to manage and process the data of digital counterpart of the system.

Shangguan [28] proposes a hierarchical model of DT for CPS design. The hierarchical high-level models that represent the DT, enable to store the whole life-cycle of products. To demonstrate the functionalities of the proposed models, an industrial robot has been presented.

Bauer [3] presents challenges and impacts of the activities of the system development and architecture design approaches that are generated by the introduction and exploitation of DTs. The authors also present proposals for tailoring and extending traditional engineering processes based on architectures to consider DT concepts.

Summing up, the DT concept is still in its initial stages, and therefore no consensus has been reached concerning several issues. A variety of different approaches can be observed both at a conceptual level as well as at the implementation level.

C. DT APPLICATIONS

This section presents some of the main applications in the literature that use the DT concept.

The DT concept proposes that real-world assets have a virtual representation connecting the physical world with the cyber world. The DT can monitor and control the physical part through the connection established between them. This connection allows the extraction of information generated by the sensors. For instance, in Smart Factories, digital twins are being used to optimize the operation and maintenance of physical assets, systems and manufacturing processes [29]. DT is being used in industries as a technology that provides a precise way of simulation from the concept to the real-world scenario where a virtual model has been developed and utilized to monitor, control, and optimize the tasks in a shop floor [30].

The DT, also can be used for improving supplier collaboration [15]. Another application for The DT is in the use for diagnostics and prognostics of some critical components involved in plant and machines [16], aircraft, manufacturing process, etc.

The DT used in the product life cycle is an innovative way for product design and manufacturing. In [31] a new method for product design, manufacturing, and service-driven by digital twin is proposed.

For instance, the DT can be used in for healthcare field [32] where simulations can be performed based on vital signs provided by sensors connected to users, such as wearables, and treatments with better accuracy can be provided. The DT for humans will have provided information such as the previous history and present condition of each person and will be helpful in predicting the occurrence of diseases. The DT will also provide the human with recommendations on how to improve their health based on the predictions [33].

D. CAR-AS-A-SERVICE

Pana [34] introduces the concept of Car-as-a-Service as a way to provide multiple services with the use of sensors, actuators and radio devices. The authors recommend the concept of using connected vehicles as service providers for passengers in the context of the smart city. For them, the CaaS's principal components are: Global Navigation Satellite System; High precision distance estimation; Radio connectivity; environmental sensors for assessing temperature, humidity pressure, etc; motion sensors, i.e., Inertial Measurement Unit (IMU) for assessing the traffic flow, the heading and roll of the vehicle, and quality of the road; a central cloud processing unit; social Networks; analytics.

One example of CaaS is car-sharing. Basically, in this kind of service, the user pays to use a car with other people. The main advantages of this service are: it can reduce the number of cars on the roads; it can save money for people that use a car only a few hours a week and decide to not own a car.

Archer [35] assumes that car-sharing schemes lead to reduced car ownership, with studies indicating that 5-15 cars are replaced for each shared car added to the fleet.

Ferrero [36] introduces a taxonomy and analyzes the different aspects of car-sharing. Most of the analyzed researches focus on two groups: a) reviews considering the technical and modeling aspects; and b) surveys that deal with the business perspectives of car-sharing. There are also challenges related to the acceptance of this kind of service by the final user.

Car-sharing is practiced in almost all European countries, the USA, Japan, China, and Australia. Mattia [37] says that it is expected to be used by approximately 12 million people by 2021. Together with this sharing approach for using cars, new services will be created to bring more comfort and safety to users such as user preferences [38] and driver monitoring.

Concerning driver monitoring, the authors in [39] reviewed Drive Monitoring Assistance Systems (DMAS) and it was figured out that the attention level of drivers is the main element for safe driving because fatigue and distraction are the dominant causes of road hazards.

In the production context, the DT has important applicability as it can help to improve production processes, find areas of congestion in production systems, adjust configurations, and enables the forecast performance of new simulated situations [4].

E. LITERATURE REVIEW ANALYSIS

Table 1 shows the most related works analyzed in this literature review.

It is possible to see that most works done in this area are focusing on the manufacturing domain. Maintenance prediction, improvement of supply collaboration, plant assembly optimization, product life-cycle and design are some of the areas that DT can be applied in manufacturing.

Layers and components of the DT vary in most of the works, but generally they have the physical world, communication, model and on the top there are the applications. Data management and analytics, simulations and HMI are also mentioned in some of the related works.

DT is also applied to other domains, such as driving assistant, healthcare, general CPS or to represent real-world products and processes.

However, most of the existing works do not consider the ISO 23247 or another standard for modeling and deploying DT. This is important for building DT systems, especially when they can grow fast as it is the case of Smart City and CaaS.

In this context, this paper proposes a definition of key-components to develop DTs with granularity. The proposed components are arranged based on the ISO 23247. They are shown and explained in the next section.

III. KEY-COMPONENTS FOR MODELING DT

In this section, the key-components for modeling DT are proposed and described with their relationship. First, the elements were organized into 4 layers defined by the ISO 23247, where it is possible to observe the responsibilities of each element. Finally, a diagram with the relations and compositions has been presented, highlighting how the elements can be arranged together to model the system.

These key-components have been proposed to build generic systems with granularity. It is possible to describe high-level services provided by assets as well as very specific components that execute a particular task, which can be part of a whole system.

A. KEY-COMPONENTS ARRANGED IN 4 LAYERS

Figure 1 presents the DT's components grouped in 4 layers defined in the ISO 23247: Observe, collect, model and learn & act. This representation shows the elements arranged by concerns, which brings an overview of how it will work in real applications.

- *Observe*: the Observe layer contains the observable elements/assets from the real world. Observable elements can be physical things such as products or equipment or nonphysical things, as for example, processes and interactions between assets of the system.
- *Collect*: elements of this layer are responsible to enable the access of data and providing ways of controlling the assets. It also contains storage capability, making it possible to save all data that comes from the assets and other information that may be generated from the system or users.
- *Model*: this is the most important part of the proposed components. The model is a virtual representation of the real world and it ideally represents the system with high fidelity.
- *Learn & Act*: the last layer contains the HMI applications that provide an interface to the users of the system. This can be via a web or desktop application, mobile application, or any other way that makes it possible the interaction between users and the system. In this layer, there are also tools & utils, which are functionalities that may be available to the users or that can be executed automatically such as event triggers.

Data flows from the observable elements up to the learn and act layer, where they are processed and analyzed. Likewise, commands can be sent back to the observable elements, changing the state of the real-world application.

B. COMPONENTS RELATIONS AND COMPOSITION

Figure 2 presents each element of the DT with its composition and with the relation between them.

- *Digital Twin*: the DT is a combination of models that represent an asset that can be observed. It can have several tools and utilities that enable to work with data, HMI that enables the interaction with final users, storage that provides a way of storing data, and communication interfaces, that allows the asset to communicate to the external world. Finally, a DT can be composed by others DTs.
- *Storage*: usually, DTs have a way of saving historical data and logs of interactions/modifications made to the asset along its lifecycle. Having the historical data of assets is essential to applications involving big data analytics and predictive maintenance. The implementation of a storage system can be done via a local file record, a local database, or a remote database hosted on a server. This is strongly related to the amount of data generated, the storage capability of the asset, and the connectivity to the external world.
- *Communication Interface*: details related to the communication such as protocols, addresses, ports and so on, must be defined to enable the communication between the real device and its DT.

TABLE 1. Table of Related Works.

Paper	DT architecture	DT composition and model	Application area
[14]	Condition information in Cloud Manufacturing	model includes a comprehensive range of manufacturing equipment information, including the basic information, the functional information and the manufacturing process information	manufacturing
[15]	a model of digital twin workshop based on physical perception data	the plant physical model, digital model (ontology) and 3D virtual model	manufacturing
[17]	framework of DT-CPPS	the cohesion of three physical portions (i.e., smart part, smart shop floor and smart manufacturing operations); the mapping and interaction of physical portions with virtual portions (i.e., virtual smart part, virtual smart shop floor and virtual smart manufacturing operations)	manufacturing in smart shop floors
[18]	open source approach for implementation based on building blocks and high-level micro-services architecture	components for the management of data, models and services	Smart Manufacturing
[2]	Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model	Physical layer, Digital layer, Cyber layer, and communication for data exchange among the three layers	manufacturing
[24]	DR-drive rapid reconfiguration on an open architecture model and open architecture of machine tools	machine level(open architecture standard platform), control level (REST-based IIoT model), system level (DT simulation, CNN based reconfiguration), Reconfigurable manufacturing systems (RMS)	manufacturing systems
[40]	architecture for a digital twin, which enables the exchange of data and information between a remote emulation or simulation and the physical twin	physical twin, local data repositories layer , an IoT Gateway layer, a layer with cloud-based data repositories and, a layer with emulation and simulation software	manufacturing system
[26]	novel security architecture	Digital twin component, Physical twin component, Protected connection between synchronization gateways, Protected connection from isolated physical twin to synchronization gateway, Production system external server, Intrusion Detection System (IDS), Security analysis service, Central access control, Protected virtual network	digital twins can work as a security enablers
[8]	uDiT Platform - distributed digital twin cooperation framework	a data centric communication middleware based on OMG (Object Management Group), DDS (Data Distribution Service), middleware and digital twin runtime engine interface, co-simulation functions based on FMI (Functional Mockup Interface), gateway functions for media and protocol conversion	Advanced driving assistant system
[5]	general framework for the Digital Twin	perception layer, communication layer, middleware layer, application layer	manufacturing
[27]	Based on the Industrial Internet of Things Open Platform Communications Unified Architecture	sense and actuate, structure and exchange unified data, services to manage and process the data of digital counterpart of the system	manufacturing
[23]	Generic architecture of CPPS based on DT	(1) physical layer, (2) network layer, (3) virtual layer, and (4) application layer	cyber-physical production system
[28]	Hierarchical Digital Twin Model Framework for Dynamic Cyber-Physical System	Virtual Interface layer, Virtual components layer, Virtual (sub-) system layer, Virtual scenarios layer	cyber physical systems
[3]	view-based architecture design using Embedded Modeling Profile	System requirements, constraints, and context; System functions and functional decomposition; Logical structuring of components; Technical realization as hardware and software parts	real-world products, entities, and processes

 – *Access Control*: security policies are one of the most important parts of every system, especially when they can control the real world. On the other hand, being able to communicate with external entities is crucial to make the system available to all involved. Therefore, details of access control must be defined to guarantee that the right parts get access to the system, especially when the DT is available via the internet. Access control measures cover user authentication and access control to the applications, which can be handled by a dedicated framework.
 – *Protocol*: IoT-based applications integrate different kinds of devices that communicate via protocols. These protocols must be mapped to the DT models and they have to be considered on the simulations and tests, to ensure increase fidelity to the real system.
 – *Application Programming Interface (API)*: APIs provide a way of accessing parts of the system by external applications. It also allows that applications developed with different technologies can still communicate to the system by using a standard protocol.
- *Tool & Utils*: a DT can have functionalities that are accessible to applications and users. These functionalities are modeled as Tool & Utils. They are callable objects that implement a set of defined instructions.

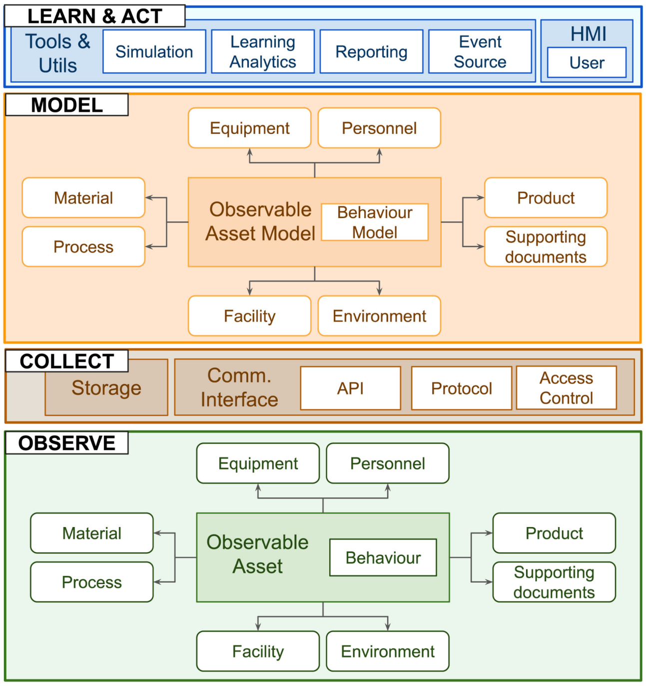

FIGURE 1. Digital twin elements arranged in 4 layers.

Tool & Utils can be used to control assets, execute simulations, run diagnostics, generate reports, etc.

- *Simulation*: simulation is part of the standard process of system development and it enables to optimize operations and predict failures [41]. It can also support planning the best layout of cooperative systems, allowing engineers to arrange the elements based on different scenarios.
- *Learning & Analytics*: monitoring real-world assets can produce a huge amount of data - Big Data - which has to be analyzed to extract meaningful information. Deep Learning can autonomously extract this information data, improving functionalities like control, optimization and health monitoring [42].
- *Reporting*: having accurate and real-time reports about the system is one key-point for enabling stakeholders to make the right decisions and improving transparency.
- *Event Source*: events can be generated to warn about something or to alert that a given condition has been met. One common example is when the temperature of a certain device is higher than expected, then an alarm will be triggered to alert about this situation. These triggers are usually defined by expert users who have knowledge about the asset.

- *HMI*: interfaces that users can visualize the DTs of the system are important to keep all stakeholders aware of the system's status. HMI can be implemented as graphical representations as for example using augmented reality techniques.
  - *User*: represents the users who will interact with the system via an HMI application. They are a very important piece since most systems are designed for users to use.
- *Observable Asset Model*: it is the representation of the real-world assets that will be observed by the system to create the DT. One observable asset can be composed of other assets or a set of assets.
  - *Behaviour*: Models such as State Charts and Timed Automata can be used to describe systems' behavior. This can help to improve the understanding of the system and allows verification of functionalities via model checking. These models can also be used for simulations, as described in the previous points.

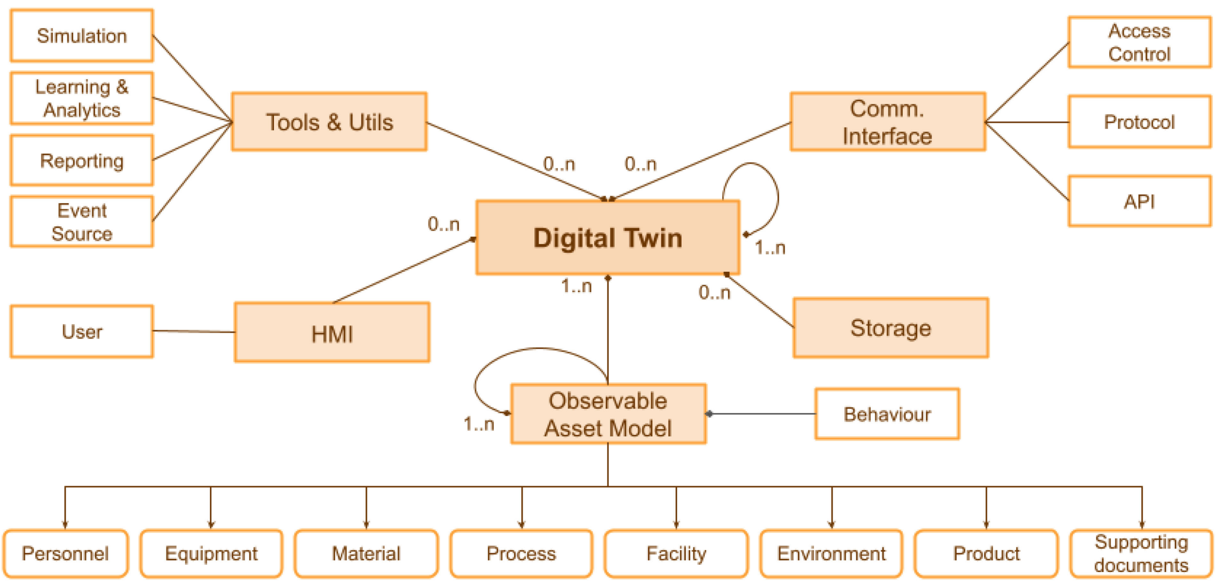

FIGURE 2. Digital twin key-components with their relationships.

IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING

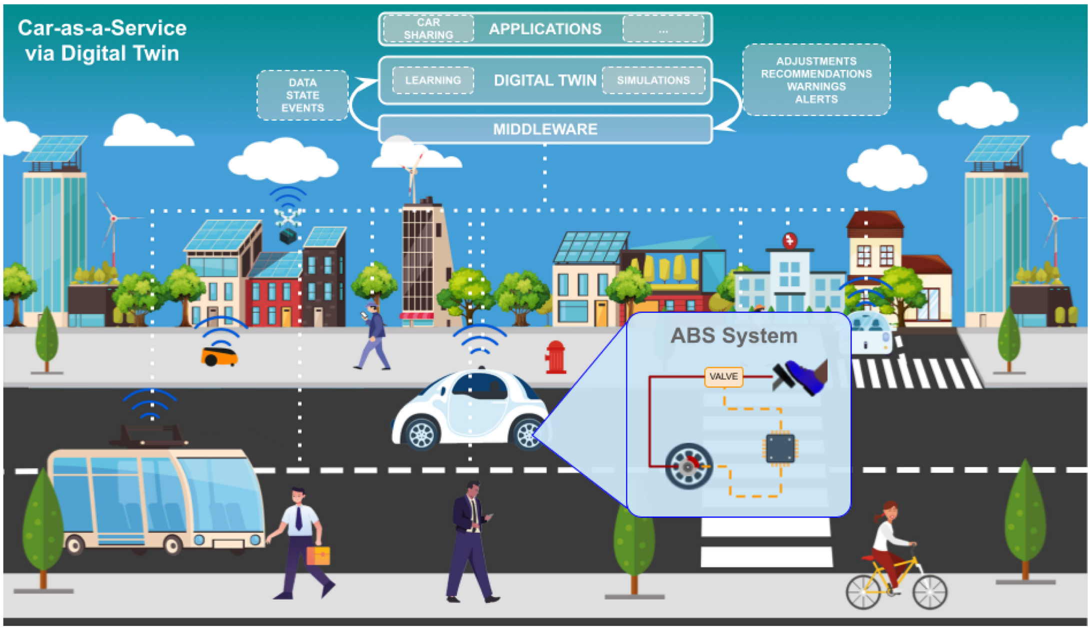

FIGURE 3. **Use case scenario.**

– *8 Sub-classes*: According to the ISO 23247, 8 classes of observable elements have been identified for manufacturing resources: Material, process, facility, environment, supporting documents, product, personnel, and equipment. Even though these elements have been initially identified for the manufacturing field, they can be applied to DT in other areas, such as in the context of CaaS.

It is possible to see that the composition of the DT can be flexible since most relations have cardinality *0..n*. This cardinality means that it is possible to have no instance of the other element (no relation) or many of them. This allows modeling a simple system with fewer components as well as a complex system with many relationshis.

Relations can be modeled by linking different files of the models. For instance, a car is modeled by combining several models for each of its parts. This combination can be done by referencing the full path of the target model. Also, models that use an open standard can be read, and details of them can be extracted, allowing to link their internal elements.

Combining models can also be done in situations where one element is represented by more models, each of them with different granularity levels. In this scenario, a new element can be created to represent the component, and all other models are linked to it.

By enabling modeling with granularity, systems can be built with different levels of detail. Challenges like ambiguity can appear when using this approach. However, having all models linked to the component they represent is still a key benefit to be considered.

To demonstrate the proposed concept, a use case has been developed and presented in the next section. It contains a simple sub-system of the ABS brake of a car as well as a more detailed one to represent the services in a smart city and CaaS.

IV. USE CASE

In this section, a use case for Car-as-a-Service has been implemented to show how the defined elements of this paper can be applied to real-world applications. It is possible to observe that the proposed elements are flexible enough to model different parts of a large application including specific parts, such as the Anti-lock braking system (ABS) as well as model services provided in a smart city, such as CaaS. This shows that the proposed concept can have different granularity levels.

A. SCENARIO DEFINITION

Together with this new way of using cars comes also a need of having new approaches of digitalization and standardization. In this context, DT can be a suitable solution for digitalization, since it provides a virtual representation of a real-world asset.

Figure 3 illustrates a scenario of a Smart City with some main elements of this context. People, cars, robots and other devices that may exist are connected via the internet to an IoT middleware. This middleware supports the main available standards and protocols which makes it possible to connect devices of different brands or generations.

Data, states, and events flow from the real-world assets into the virtual world to the respective twin. Triggers can be executed every time new data is received and after processing this data, the DT can send adjustments/recommendations or alerts to the real world. By using this approach it is also possible to store all lifecycle of each individual element as well as data related to the interaction between two or more objects. This allows the system to evaluate and improve future relationships between the entities of the system.

In this kind of scenario, there may exist different levels of granularity in the whole system. For instance, DT can be used to model technical details of cars since its design [4]

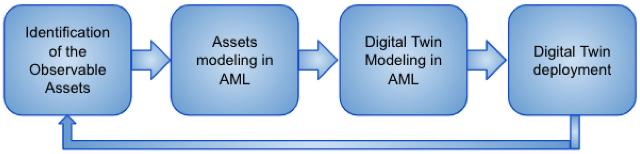

FIGURE 4. **Steps for modeling and deployment of DT.**

and it can be also used to represent services provided by the objects in a smart city.

Applications like Car-sharing can be created on top of the digital twins and they can take advantage of accessing high-level objects without handling low-level issues like different protocols for each device. The applications can also offer customized services based on the previous data from the DT that is stored during the whole lifecycle of the asset.

However, to have such a generic and compatible system it is necessary to model the elements in a common language that can be understood across the entire system.

B. DESIGN AND IMPLEMENTATION OF DIGITAL TWINS

Different approaches related to modeling DT can be found in the literature [43] [44]. It is possible to identify similarities in the way these works understand the modeling and deployment of a DT. Figure 4 shows the main steps to model and deploy DT, and a description is given as follow:

- *Identification of the Observable Assets*: the first step is to identify all elements/assets that will be observed in the system. It is not limited to physical devices, but it can be anything that is observable such as processes and interactions between assets.
- *Assets modeling in AML*: then, all observable assets from the real world are modeled with their attributes and the relationship between them in AutomationML.
- *Digital twin modeling in AML*: based on the model of the previous step, the digital representations are created in this step. Communication interfaces (e.g., IP and port) and storage details are defined in this step. Tools and utilities like events and triggers can also be modeled to enable the DT to send adjustments and feedback to real-world assets.
- *Deployment*: finally, the deployment is based on the already created models. Depending on the model format, it is possible to automatically extract information from them [20], and generate communication interfaces for the integration between the real-world asset and its DT.

Following the first and the second steps, which is to identify the assets of the real world and their relationships, Figure 5 illustrates these elements and the connections between them. The two main parts of the system are the real-world, in green, and the cyber world, in blue.

Users keep their personal data updated with information that is provided by their smartphones. Some of the smartphone manufacturers already provide a platform to access fitness information about the device's user. For more accuracy, wearables can be used to measure vital signals and triggers can be created to notify users when any sign is out of the normal range.

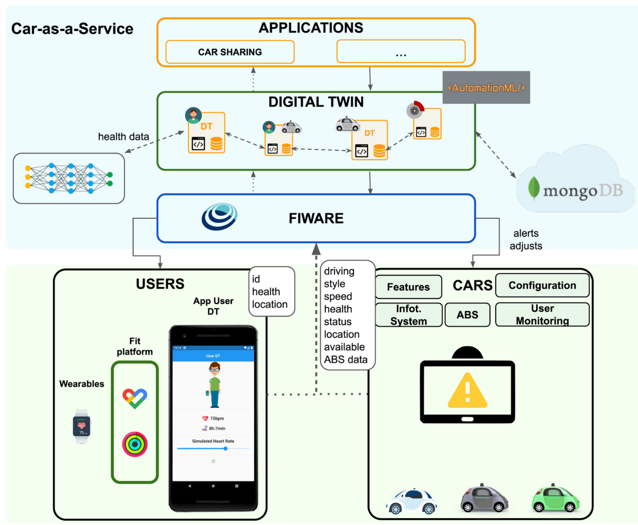

FIGURE 5. **Elements considered for the use case.**

In parallel, cars continuously send data related to the driving and the current user, as well as information about their own working condition.

In this scenario, it is possible to see that applications may have different levels of granularity since high-level services can be model together with specific systems inside the car such as the ABS system.

The function of the ABS is to prevent that the wheels are locked when the driver is braking based on the calculated slip rate. Figure 6 shows a timed automata model for the ABS. When the velocity is zero, the state goes to end with torqueABS zero as well. It means that the valves will open and the system will not brake. If velocity is greater than zero, the slip rate is calculated, and based on this calculation, the state goes to the end whether with or without torque.

This simplified timed automata model illustrates how it is possible to model a system based on the behavioral point of view without giving any timing information related to the ABS system. A system or an asset can be model based on different perspectives, depending on its functionalities.

The main model, which represents all digital twins has been done with AutomationML. Figure 7 shows part of the model with the observable elements of this use case.

It is possible to observe that the three main parts are the ABS System, User, and Car. The ABSSystem is an observable element that contains behavior that is linked to the timed automata. Likewise, the ABS is contained in the Car, which has other properties and its own digital representation. This shows how granularity can be achieved through this approach.

The possibility of creating links to external models is important in this kind of system since each part of it can have different roles and functionalities that are represented also in distinct ways.

After users and cars are modeled, communication interfaces are generated automatically via a script written in Python. This script extracts all entities from the model with their attributes

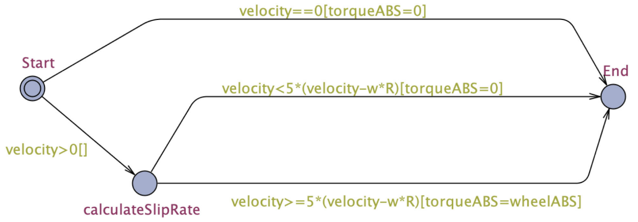

FIGURE 6. **Simplified timed automata of an ABS system.**

and configures the communication which in this case study is made through a middleware IoT called FIWARE [45].

C. ADOPTED TECHNOLOGIES

For this use case, several technologies have been used, from the modeling to the implementation.

For modeling, the AutomationML has been selected since it is an established standard in the industry. AML is an XML-based neutral data format for the storage and exchange of engineering information. It also allows reusability and interoperability between different applications and languages. AML files allow applications to extract its information since its content is saved as text in the XML format.

To implement the infrastructure necessary for connecting all assets, the middleware FIWARE has been used. All elements from the real world are connected via this middleware, and each element is represented as an entity that can be manipulated via HTTP requests.

- POST: used to create a new entity.
- PUT & PATCH: used to update an existing entity.
- GET: used to retrieve an entity.
- DELETE: used to delete a specific entity.

An important feature of FIWARE is the possibility of subscribing to updates on a specific entity. Using this functionality, a script for keeping the Digital Twins updated was implemented. Every time that a new update on the user's data was received, this data was evaluated and a prediction of heart attack was done via a neural network.

CaaS applications may also use users' data to provide custom service for each person. Therefore, a mobile application has been developed in Flutter/Dart to access the user's fitness data through the Google Fit platform. In this application, the user has to login with his/her Google account and grant access to the fitness data. It is also required that the user enter their personal information to keep the profile always updated. This app keeps the digital twin of the user updated with the most recent data available. Through an API, the app requests the health data to the Google Fit platform every period of time. This period can be configured based on the kind of user or on the information that is being requested.

For simulation purposes, an interface to external systems has been developed to allow the communication between the car and its digital twin. Through this interface, it is possible to send alerts or change some configuration of the car such as the max speed limit.

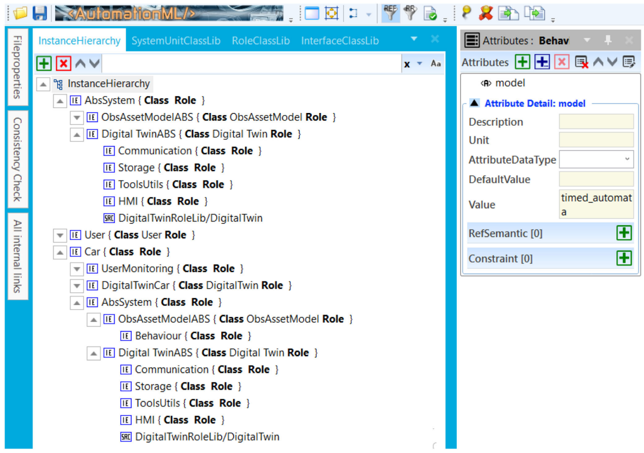

FIGURE 7. **AutomationML ABS system.**

D. SUMARY OF THE USE CASE

This use case showed how the proposed concept can be applied to a real scenario and it was possible to identify the advantages of using the concept of Digital Twin to enable and support Car-as-a-Service. One important advantage is that all entities have a virtual representation that can be accessed via the middleware. Another benefit of using this approach is the possibility of tracking historical data from all entities such as cars, users and the relations between users and cars, making it possible to improve the user experience in futures rides.

It has also shown how different models need to be integrated to form a DT of the whole observable asset. Have all models linked to the main DT can help to have an overall overview of all parts together.

Also in this use case, automatic integration of these models was not been developed, therefore, the model checking of the ABS timed automata would have to be executed manually.

V. DISCUSSION

A concept of key-components for modeling DT has been proposed and a use case for supporting CaaS by DT was developed to demonstrate how it can be applied in a real scenario.

A DT can cover not just a physical device, but it can represent also nonphysical things such as processes, interactions, methods, and so on. This increases the need of modeling the behavior of the system, which improves the simulations and tests.

In an ideal scenario, the simulations can be performed based on mocks objects that reproduce the behavior of the real-world objects/assets. In some cases where not all assets are available, simulations can also be implemented by mixing mocks and real-world objects.

By introducing different clocks in timed automata, deadlines can be proved, which is important for hard and soft real-time systems. By using this kind of model it is also possible to perform model checking. This can be used to find possible failures in the system, especially when some parts are added or modified.

Similar to most model-based approaches, consistency is an important aspect to be considered while building a system. One common example is data type matching, which many tools already provide. In this concept, this aspect has not been approached in-depth, relying on existing tools such as AutomationML. However, new rules can be developed to cover the aspects related to consistency.

AutomationML is widely used in industry for building engineering data models, and this makes it an important candidate for modeling technology. It offers a flexible way of modeling since it is possible to create classes and instances of the system's components. However, for real-world applications, such as smart cities, it can bring some problems related to performance or/and usability since the number of components and data is always growing.

Modeling every single element of the system can be infeasible due to the amount of time needed for that. Therefore, creating representations of a set of elements that can be instantiated as objects can be one solution for this problem.

VI. CONCLUSION

Technology innovations are the base and stimulate new business models such as smart cities and car-as-a-service. CaaS will change the way people see and use cars, especially when autonomous cars will be fully available in the market. This kind of application brings new challenges such as the digitalization of real-world assets.

In this context, this paper proposes the definition of the key-components for building digital twins. These elements have been arranged in four layers defined by the ISO 23247 to better identify the functionalities/responsibilities of each proposed component.

DT modeling has to handle the integration of different models that represent assets in different aspects. This paper addressed this by providing an architecture that supports several models linked to one single asset.

The concept has been applied to a use case of CaaS in smart cities. It was possible to demonstrate that systems can be modeled with different levels of granularity, such as high-level services provided by cars as well as low-level systems like ABS brake of a vehicle.

There are still open challenges that will be addressed in future works. Integration between models is an important challenge that has to be handled since it can bring ambiguously and consistency issues. Current modeling tools may not be scalable enough for real-world applications, especially for the ones that have many components. Therefore, new approaches that are scalable and provide a query mechanism are needed. A use case with 3D representations for end-users interactions is planned. Also, components for security will be added, such as blockchain-based solutions.

REFERENCES

[1] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: A review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[2] S. Aheleroff, X. Xu, R. Y. Zhong, and Y. Lu, "Digital twin as a service (DTaaS) in industry 4.0: An architecture reference model," *Adv. Eng. Informat.*, vol. 47, 2021, Art. no. 101225.

[3] T. Bauer, P. O. Antonino, and T. Kuhn, "Towards architecting digital twin-pervaded systems," in *Proc. IEEE/ACM Int. Workshop Softw. Eng. Syst.-Syst. Workshop Distrib. Softw. Development, Softw. Ecosyst. Syst. Syst.*, 2019, pp. 66–69.

[4] R. B. Roy *et al.*, "Digital twin: Current scenario and a case study on a manufacturing process," *Int. J. Adv. Manufacturing Technol.*, vol. 107, no. 9, pp. 3691–3714, 2020.

[5] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020.

[6] R. Petrasch and R. Hentschke, "Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method," in *Proc. Int. Joint Conf. Comput. Sci. Softw. Eng.*, 2016, pp. 1–5.

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

[8] S. Yun, J.-H. Park, and W.-T. Kim, "Data-centric middleware based digital twin platform for dependable cyber-physical systems," in *Proc. Int. Conf. Ubiquitous Future Netw.*, 2017, pp. 922–926.

[9] ISO, "Automation systems and integration digital twin framework for manufacturing," *Int. Organization Standardization, Standard ISO 23247–1:2020*, 2020. [Online]. Available: https://www.iso.org/standard/75066.html

[10] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," URL http://www. apriso.com, 2014.

[11] M. Shafto *et al.*, "Modeling, simulation, information technology & processing roadmap," *Technol. Area*, vol. 11, pp. 1–38, Nov. 2012.

[12] 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.

[13] 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. VIII Brazilian Symp. Comput. Syst. Eng.*, 2018, pp. 154–159.

[14] W. Xu, J. Yu, Z. Zhou, Y. Xie, D. T. Pham, and C. Ji, "Dynamic modeling of manufacturing equipment capability using condition information in cloud manufacturing," *J. Manufacturing Sci. Eng.*, vol. 137, no. 4, 2015, Art. no. 040907.

[15] Q. Zhang, X. Zhang, W. Xu, A. Liu, Z. Zhou, and D. T. Pham, "Modeling of digital twin workshop based on perception data," in *Proc. Int. Conf. Intell. Robot. Appl.*, 2017, pp. 3–14.

[16] C. Zhang, W. Xu, J. Liu, Z. Liu, Z. Zhou, and D. T. Pham, "A reconfigurable modeling approach for digital twin-based manufacturing system," *Procedia CIRP*, vol. 83, pp. 118–125, 2019.

[17] K. Ding, F. T. 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. Production Res.*, vol. 57, no. 20, pp. 6315–6334, 2019.

[18] V. Damjanovic-Behrendt and W. Behrendt, "An open source approach to the design and implementation of digital twins for smart manufacturing," *Int. J. Comput. Integrated Manuf.*, vol. 32, no. 4–5, pp. 366–384, 2019.

[19] D. Rovere, P. Pedrazzoli, G. dal Maso, M. Alge, and M. Ciavotta, "A centralized support infrastructure (CSI) to manage CPS digital twin, towards the synchronization between CPS deployed on the shopfloor and their digital representation," in *The Digital Shopfloor Industrial Automation in the Industry 4.0 Era: Performance Analysis and Applications*. Denmark: River Publishers, 2019.

[20] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with automationml and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.

[21] S. Sierla, V. Kyrki, P. Aarnio, and V. Vyatkin, "Automatic assembly planning based on digital product descriptions," *Comput. Ind.*, vol. 97, pp. 34–46, 2018.

[22] J. Um, S. Weyer, and F. Quint, "Plug-and-simulate within modular assembly line enabled by digital twins and the use of automationml," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 904–15 909, 2017.

[23] H. Zhang, Q. Yan, and Z. Wen, "Information modeling for cyber-physical production system based on digital twin and automationml," *Int. J. Adv. Manuf. Technol.*, vol. 107, pp. 1–19, 2020.

[24] J. Leng *et al.*, "Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model," *Robot. Comput.-Integrated Manuf.*, vol. 63, 2020, Art. no. 101895.

[25] A. Redelinghuys, A. Basson, and K. Kruger, "A six-layer architecture for the digital twin: A manufacturing case study implementation," *J. Intell. Manuf.*, vol. 31, pp. 1–20, 2019.

[26] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 669–680, Jan. 2019.
[27] 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. Consumer Electron.*, 2019, pp. 1–2.
[28] D. Shangguan, L. Chen, and J. Ding, "A hierarchical digital twin model framework for dynamic cyber-physical system design," in *Proc. Int. Conf. Mechatronics Robot. Eng.*, 2019, pp. 123–129.
[29] F. Ameri and R. Sabbagh, "Digital factories for capability modeling and visualization," in *Proc. IFIP Int. Conf. Adv. Production Manage. Syst.*, 2016, pp. 69–78.
[30] J. Vachálek, L. Bartalský, O. Rovný, D. Šišmišová, M. Morháč, and M. Lokšík, "The digital twin of an industrial production line within the industry 4.0 concept," in *Proc. Int. Conf. Process Control*, 2017, pp. 258–262.
[31] 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, pp. 1–14, 2017.
[32] 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.
[33] K. Bruynseels, F. Santoni de Sio, and J. van den Hoven, "Digital twins in health care: Ethical implications of an emerging engineering paradigm," *Front. Genet.*, vol. 9, p. 31, 2018.
[34] C. Pana, S. Severi, M. Raffero, C. Dannheim, and G. Abreu, "The newest road revolution: Car as a service," in *Proc. AmE 2017-Automotive Electron.; 8th GMM-Symp.*, 2017, pp. 1–4.
[35] G. Archer and B. Bondorova, "Does sharing cars really reduce car use," 2017. [Online]. Available: https://www.transportenvironment.org/discover/does-sharing-cars-really-reduce-car-use
[36] F. Ferrero, G. Perboli, M. Rosano, and A. Vesco, "Car-sharing services: An annotated review," *Sustainable Cities Soc.*, vol. 37, pp. 501–518, 2018.
[37] G. Mattia, R. G. Mugion, and L. Principato, "Shared mobility as a driver for sustainable consumptions: The intention to re-use free-floating car sharing," *J. Cleaner Prod.*, vol. 237, 2019, Art. no. 117404.
[38] F. Bardhi and G. M. Eckhardt, "Access-based consumption: The case of car sharing," *J. Consumer Res.*, vol. 39, no. 4, pp. 881–898, 2012.
[39] M. Q. Khan and S. Lee, "A comprehensive survey of driving monitoring and assistance systems," *Sensors*, vol. 19, no. 11, p. 2574, 2019.
[40] A. Redelinghuys, A. Basson, and K. Kruger, "A six-layer digital twin architecture for a manufacturing cell," in *Proc. Int. Workshop Serv. Orientation Holonic Multi-Agent Manuf.*, 2018, pp. 412–423.
[41] S. Boschert and R. Rosen, "Digital twin the simulation aspect, " in, *Mechatronic Futures*. Berlin, Germany: Springer, 2016.
[42] J. Lee, M. Azamfar, J. Singh, and S. Siahpour, "Integration of digital twin and deep learning in cyber-physical systems: Towards smart manufacturing," *IET Collaborative Intell. Manuf.*, vol. 2, no. 1, pp. 34–36, 2020.
[43] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020.
[44] G. N. Schroeder, C. Steinmetz, R. N. Rodrigues, R. V. B. Henriques, A. Rettberg, and C. E. Pereira, "A methodology for digital twin modeling and deployment for industry 4.0," *Proc. IEEE*, vol. 109, no. 4, pp. 556–567, Apr. 2021.
[45] 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.

CHARLES STEINMETZ received the BS degree in information systems from Franciscan University, Santa Maria, Brazil, in 2015, and the MSc degree in computer science from the Federal University of Rio Grande do Sul, Porto Alegre, Brazil, in 2018. He is currently working toward the PhD degree in computer science with the University of Oldenburg, Oldenburg, Germany. He is also a research assistant with the University of Applied Science Hamm-Lippstadt, Lippstadt, Germany. His research interests include system architectures, human machine interface technologies for embedded devices, Internet of Things, and Digital Twin.

GREYCE SCHROEDER received the BS degree in computer engineering from the Fundao Universidade Federal do Rio Grande, Rio Grande, Brazil, in 2005, the MSc degree in electrical engineering form the Universidade Estadual de Campinas (Unicamp), Campinas, Brazil, in 2007, and the DEng degree from the Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, in 2018. Her general research interests include industrial automation, robotics, human machine interface technologies, Cyber Physical Systems, Internet of Things, Industry 4.0, and Digital Twin.

RICARDO NAGEL RODRIGUES received the bachelor's degree in electric engineering from Instituto Nacional de Telecomunicaes in 2003, the master's degree in electric engineering from Universidade Estadual de Campinas in 2006, and the PhD degree in computer sciences from the State University of New York at Buffalo in 2011. His research interests include computer science, focusing on computer vision, acting on the following subjects: industrial computer vision, deep neural networks, transfer learning, and pattern recognition. He is currently a full professor of automation and computer engineering with the Federal University of Rio Grande, Brazil.

ACHIM RETTBERG received the MS (Dipl.-Inform.) degree in computer science and economics in 1997 and the PhD degree in 2006 from the University of Paderborn, Germany. Since December 2017, has been a professor of human machine interface technologies with the University of Applied Science Hamm/Lippstadt. Prior to this he was an extraordinary professor with Carl von Ossietzky University Oldenburg. From 2007 to 2008 he was a postdoc with C-LAB and managed EU-Projects. He is the founder and general chair of the International Embedded Systems Symposium (IESS). Since April 2012, he has been the chair of the IFIP working group 10.2 for Embedded Systems and since October 2019, he has been the chair of IFIP TC 10 Computer Systems Technology. His general research interests include real-time systems, HW/SW architectures, design methods, optimization for embedded systems and architectures, and for human machine interface technologies for embedded devices.

CARLOS EDUARDO PEREIRA received the BS degree in electrical engineering and the MSc degree in computer science from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 1987 and 1990, respectively, and the PhD degree in electrical engineering from the University of Stuttgart, Stuttgart, Germany, in 1995. He is currently a full professor with UFRGS and the director of operations of EMBRAPII, an innovation agency in Brazil. He is the deputy-editor-in-chief of the *IFAC Journal of Systems and Control* and an associate editor for the *Journals Control Engineering Practice and Annual Reviews in Control*, all from Elsevier. He is also the vice-president for the Technical Activities of the International Federation on Automatic Control (IFAC). He was the recipient of the Friedrich Wilhelm Bessel Research Award from the Alexander von Humboldt Foundation Germany, in 2012.