Digital Twins: Review and Challenges

Maria G. Juarez, Vicente J. Botti, Adriana S. Giret

Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Valencia, 46022, Spain

Abstract

With the arises of Industry 4.0, numerous concepts have emerged, one of the main concepts is the Digital Twin (DT). DT is being widely used nowadays, however, as there are several uses in the existing literature, the understanding of the concept and its functioning can be diffuse. The main goal of this paper is to provide a review of the existing literature, to clarify the concept, operation and main characteristics of DT. Also, to introduce the most current operating, communication and usage trends related to this technology. And present the performance of the synergy between DT and MAS (Multi-Agent System) technologies, through a computer science approach.

Keywords: Digital Twin, Manufacturing, Literature Review, Multi-Agent System, Ontologies

1 Introduction

The Industry 4.0 is trending nowadays in terms of smart manufacturing and information sharing. One of the main characteristics of this new industrial wave is the introduction of two main concepts: Internet of Things (IoT) and Services (IoS), in all the industry fields. According to the experts this revolution will excel the previous ones, cause its presents better performance when it comes to adaptability, interconnection, exercising judgement, productiveness and effectiveness [46, 71]. The Industry 4.0 presents many advantages for the manufacturing industry, admit smooth unification between intertwined smart elements that belong to a shop floor, also permit simultaneous observation and manage of gadgets and cyber-physical components through network frameworks, and consequently allows an uncomplicated fusion and coordination among the physical and the digital world [64, 54, 50].

The fourth industrial revolution also introduces a new concept, calls Digital Twin (DT), it's a digital substitute of a real object compose of several models (multi-physical, behavioral, etc.) and data exchange functions, creating intelligent gadgets that act in the nucleus of the IoT and IoS [80, 64]. Currently, it's widely used as it acts as a catwalk between the real and the digital world, supplying smart manufacturing companies with a novel way of carrying out smart fabrication and accuracy control [69].

Based on a comprehensive literature research, this paper presents all the necessary concepts to understand the functioning of DTs. The main goal is to provide a holistic overview of the principal concepts, functioning, and main characteristics of this technology. As well as present the main trends of operating, communication and use based on various case studies.

The article is organized as listed below. Section 2 deals with the concepts, operation related aspects, main characteristics and evaluation tests for DT. Section 3 shows the main data handling trends. Section 4 introduces the main communication trends. Section 5 exposes the methodology of the literature classification used for the main usage trends and presents the outcome of it. Section 6 proposes some challenges ahead and concluding remarks.

2 Definition and Main Concepts

2.1 Definitions

There are numerous definitions of DT, the first known definition was presented by NASA, the DT is a scale, physic, unified, stochastic simulation, which employs the best accessible models (physical, behavioral, etc), as well as the updated information to emulate the life cycle of its physical twin (flying device) [83], an almost complete list of different definitions of the digital twin concept can be found at [64]. Most of the definitions textually mention the use of physical, electronic, and structural models, to simulate the life cycle of the element of interest. Since the given definition by Schluse in his work [80], arises the idea to add intelligence to digital twins and use them to make forecasts of behavior and failures in systems.

The definition given by Lee [55], it's the first one that allows the coupled model of the element to be simulated to execute on a cloud platform. Majumdar in his work [60], defines the only structural model which includes quantitative data to properly work. Schroeder [82], is the first one to mention in his definition the use of digital twin technology with CPS (Cyber-Physical Systems).

A widespread concept of DT that has been recognized and employ by numerous people so far is given in [49], a DT is a scale, physic, unified, stochastic simulation of an as-built system, permitted by the use of a Digital Thread, which employs the best accessible models (physical, behavioral, etc), as well as the updated information to emulate the life cycle, actions and operation of its real twin [64]. This concept is the most widely used because it's a broad and complete definition that includes all the main characteristics and elements that are part of a DT, and the main characteristics that rule its performance. It's also important to highlight that this definition, unlike the first given by NASA, can be used in different areas of development, since its main objective is not reduced to reflecting the life of an air vehicle, but rather reflecting the life cycle of any element, product or system that works as its physical twin.

2.2 Main Components

According to Grieves [37, 70], DT contains the following principal components: (i) physical objects in the Physical World, ii) digital objects in the Digital World, and (iii) the set of connections that ties the digital and physical elements with each other.

2.2.1 Physical World

In agreement with Zheng [101], the physical world is an intricate, varying, dynamical, manufacturing atmosphere, which containing the following elements: users and operators, assets, machinery, goods, specifics regulations and atmosphere. The assets summarizes all the necessary elements for the manufacturing production, including those related to production, data capture, and the necessary software devices. All these elements have their own place, but they need to linked trough IOT systems.

The physical world is composed by two main elements:

1. Devices: the physical twins from which the digital twins are intended to be created.
2. Sensors: elements that are physically connected to the devices, and through which data and information are obtained, once the sensor obtain the data, it sends it to the Physical World where is processed. The most widely spread used sensors are: PLC (Programmable Logic Controller), RFID (Radio Frequency Identification), QR code (Quick Response Code), among others.

2.2.2 Digital World

Following with Zheng's work, the digital world contains two parts:

1. The virtual environment platform (VMP), constructs an integrated 3D digital model to execute apps, and on the same time allow executing actions to prove the functioning of diverse algorithms. There are numerous connections among the VMP and the DTs, cause VMP offers different necessary models to the development and performance of the DTs.
2. Digital Twins, which mirror their physical entities life course and allow multiples operations (control, prediction, etc).

2.2.3 Connections among the Physical and the Digital World

Connections between real and virtual spaces are different depending on the development methodology that each author uses, the most used technologies will be discussed in section 4.

2.3 Terms

According to the work of Grieves and Vickers [38, 15], the next expressions conceptualize what a DT means, (1) Digital Twin Prototype (DTP): A DTP enumerate the actions to produce a specific object, for example the list of actions necessary to create a product; (2) Digital Twin Instance (DTI): A DTI explains the elements that are part of a certain instance of an object, for example the names and specifications of the materials needed to produce this object; besides includes the real-time situation of the sensors linked with the object; each instance of the object will have their own DTI and will be created using an only DTP; (3) Digital Twin Aggregate (DTA): A DTA is an addition of numerous DTIs, this aggregation enable obtain information of the objects.

2.4 Digital Twin Classes

Cardin [20] in his work states that: The concept of DT is starting to be used in smart manufacturing specifically in management frameworks. And it has become an essential component to take decisions, resources management, with the capacity to check the status, modify the behavior and realize predictions of the real object. The following ones are the principal classes of a DT:

1. DT of products, scanning, patterning and management data information related to the cycle of life of a specific product. The original concept was introduced by NASA to control the life-cycle of a flying device. The information perceived by the sensors and the information nearly in real time nourish the distinct simulations which mirror the behavior of the flying device [97, 83].
2. DT of systems, reflecting and predicting the conduct during the life-cycle of a system. This class is used primarily in fields such as medical care, fabrication, and logistic-planning, as it can be used to achieve multiple goals, for example supervising the systems to detect failures of

the real twin, and from these failures perform predictive maintenance.

2.5 Digital Twin Categories

As proposed by Borangiu [15], it is possible to affirm that the application of the DTs in the Industry 4.0 can divide in the following types:

1. Plain gadget models; include two main groups of data: (a) current value group, data-information obtained by the sensors of the object, this data update the DT features; (b) a set of expectations values, that want to be obtained by the gadget [43].

2. Embedded DTs (EDT); they participate in all operations that include their twins, for example the simulations models from a specific object or the job scheduling from a system. The interactions among the physical and the digital world occurs through a bidirectional connection, the data from the real objects are obtained by the sensors, which allow take smart decisions through the DT [65].

3. Networked twins; networking gives the capacity to each integrated EDTs to connect with the different EDTs of the atmosphere, allowing communication among each others. This can be really helpful in smart manufacturing environments.

2.6 Level of Integration

According to Kritizinger [50], as claimed into the literature related to the DT conceptualization it's possible to infer that digital twins are the digital copies of their physical twins. In this literature the expressions: Digital Model, Digital Shadow and Digital Twin are used on several occasions and as synonyms, nevertheless these expressions vary in the level of integration of the data among the real and the digital twin. Some instances of DTs are simulated and obtain their data from files entered by the operators, however others obtain their data directly from their physical twins. Thus, Kritizinger propose a classification of DTs, divided into the following kinds, based on the level of integration.

Digital Model

It's a virtual portrayal of a simulated or real object that doesn't utilize any automated information interchange among the real and the virtual subject. The virtual portrayal may contain a rough description of the real object, this means it can contain some models (physic, behavior, etc.) of the specific object, as long it doesn't use automated information interchange. Data can be used for creating models and simulations as long as it's entered manually. If there is a change in the real object doesn't directly affect the virtual object and vice versa. The flow of data between the twins can be seen at Fig. 1.

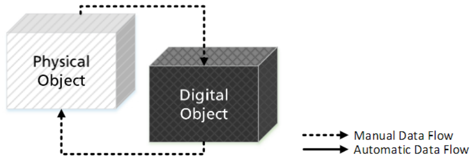

Fig. 1. Flow of Data Digital Model

Digital Shadow

Instead, if exists an automatic unidirectional information interchange among the status of the real object and the virtual object, this will be call a Digital Shadow. So if occurs an alteration in the state of the real object, it will directly lead to an update in the virtual object, however not on the contrary. The flow of data between the twins can be viewed at Fig. 2.

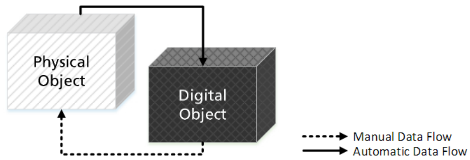

Fig. 2. Flow of Data Digital Shadow

Digital Twin

But besides if exists information interchange among a real object and a virtual object in a bidirectional way, this will call a Digital Twin. Which gives the DT the ability to be an instance of its real twin and to be used for managing its life cycle. If a change occurs either in the real or virtual object, this will directly affect the other. The flow of data between the twins can be seen at Fig. 3.

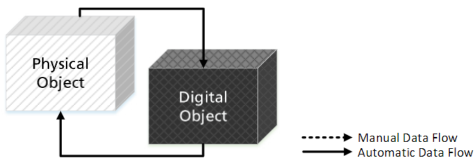

Fig. 3. Flow of Data Digital Twin

2.7 Main Characteristics

As proposed by Tao [87], the main characteristics that a DT presents are the following:

1. Real-time reflection: There are two worlds inside a DT, the physical and the digital world. The digital one mirror the real state of the physical world, and based on the exchange of data between the worlds it's possible that it's synchronized.

2. Communication and confluence: It presents the following aspects. (i) Communication and confluence in the physical world. The digital twin is a completed and unified system which include the information produced in the entire life-cycle of the real twin, this allows that the information generated can be fed back. (ii) Communication and confluence among the stored information and the current information. The information comes from different sources which lead to it being more reliable and complete which allows a better use of it. (iii) Communication and confluence among the physical and the real world. There is a bidirectional relationship between these worlds, which allows information exchange between them [30].

3. Self-evolution: The DT has the capacity to refresh and modified the true-time information, which produces a direct, consecutive and positive change in the models it contains by comparing current information with that presented in the physical world [90].

2.8 Grieves' Tests of Virtuality (GTV)

According to Grieves [38], when creating virtual systems is necessary to evaluate them, to determinate if these systems have the capacity to mirror their physical counterparts. To realize these evaluations Grieves suggest a serial of trials, named the Grieves' Tests of Virtuality (GTV), these test are based on the Turing intelligence trials, created for Alan Turing in 1950 [91] called the 'Imitation Game'.

The assumption of the GTV trial is comparable to the trials creating by Turing, a real person acts as the viewer, the viewer is located inside a room that has two separate screens, one screen presents the true state of the real system, while the other shows the DT status. The tests to run are: sensory visual, performance and reflectivity.

Sensory Visual Test

The viewer demands any movement of three-dimensional nature either in the real system or in the DT, for example rotation or assembly of the pieces of the system. If the viewer can't differentiate among the real system and the digital twin, the test has been approved.

Performance Test

The viewer requires any action to be executed either on the real system or on the DT. For example open or close an industrial valve. If the viewer can't differentiate between the real system and the digital twin, the Performance Test has been approved.

Reflectivity Test

The viewer requests to check the state of any element of the system. For example, the viewer can request to observe the altitude presented by a flying twin. If the viewer obtain the same data from both the physical and the digital twin, the Reflectivity Test has been approved.

2.9 Modeling DTs

As proposed by Tao [88], a critical question regarding DT technology, is if there is a generic formula to create a feasible DT model. The author concludes that there isn't a generic way to build a DT model. Each author utilizes different methods, methodologies and modelings tools, the most widely used data modeling tools will be addressed in the next section.

3 Data Management Trends

3.1 Data Modeling

Schroeder in his work [82] states that: a DT reflects the entire life-cycle of a specific object. The DT includes various models and information from different sources. This created a matter, the necessity to create a model which contain the different necessary models to manage the DT information. To solve this matter Schroeder [82] proposes to use the following concepts:

1. **XML**: It's an extensive markup language which determinate a group of regulations to encrypt information in a way that can be read by both people and computers. This language can detail multiple kinds of information and it has been widely used with many purposes [61]. For example, in [21] the data-information standardized for PPR (product, process and resource) employs XML, this language is used to define and develop a PLM integrator (product life cycle management) that allow interchange of PPR information between heterogeneous commercial PLM systems and other systems. Other example can be seen in [100] where the Extensible Application Markup Language (XAML) based on XML, is used in case study which goal is to optimize the production of a line of hollow glass, XAML is employed to interchange reference models, cause it saves data related to different paradigms and enable model components from various sources as information objects embodying various properties.

2. **Standard for the Exchange of Product Model Data (STEP)** [68]. It's controlled to the regulation ISO 10303, it's a group of rules determining a strong methodology which details the information related to a specific product through its life-cycle. STEP has been frequently-used for various purposes: CAD (computer-aided design), and PLM management [85, 98]. STEP also saves data product, however not the data to realize object supervision and engineering control.

3. **Computer Aided Engineering Exchange (CAEX)** [67]: It's a neutral information format created from XML, which details physical or digital objects into information objects. This format is helpful not just for the interchange of data-information at industrial levels, but also for the interchange of data-information at multiple local levels. It has been used for production line control and analysis, besides it can be used for modeling industrial factories and as a bridge to transport data inside a production chain. A concrete example of the use of CAEX to model DTs is presented by Schroeder in his work [82], where CAEX is used together with AutomationML to model the sensors and actuators of an industrial valve, therefore it acts as a data source.

Another concept that can be used to resolve these problems is **Ontology**, a clear definition can be seen in [31], based on Negri [64] scientific literature research, to take a full advantage of the capacities of CPS (Cyber Physical System) and IoT, an appropriate information model should be used, like an ontology [31, 63], an ontology is a clear, grammatical and proper definition of a specific concept that belongs to a domain [39]. It's a central semantic formal naming that offers intelligence to a CPS [56] and collaborates to achieve fusion and transmission of large amounts of detected data [16, 42].

Regarding the manufacturing industry Garretti [31] states: For the manufacturing system area, one of the principal problems resides in the multiple configurations that a manufacturing system can have. Because of this, it hasn't been possible to create a generic procedure to design and manage manufacturing systems, since most systems have a different end purpose. To contribute to solving this predicament Negri [63], exposes a review of the existing literature related to ontologies and manufacturing systems, presents the requirements that the ontologies must achieve to be used in this area and then selects the most appropriate ontologies. The selected ontologies were: OIL, DAML (DAML + OIL and DAML-L), OWL and its sub-languages: OWL Lite, OWL DL, C -OWL, OWL-Eu, OWL-E, since they fulfill all the necessary conditions. A clear example of the use of ontologies can be see in [3], where the author create a Cloud-Based Cyber-Physical System Architecture (C2PS) and uses an ontology to represent the characteristics of the physical twins and to allow the critical understanding of the low level messages and required control actions between the DTs of the sytem. Another example can be seen in [62], where the author presents a synergy of the concepts addressed in the previous paragraphs, the author offers to model particular features and behaviors of a system, through FMU modules (Functional Mock-up Units) that simulate different behaviors (e.g. energy consumption, availability), to achieve this the author uses: XML to represent the internal variables of each module, AML (Automation Markup Language) to develop the semantic data model, and ontologies which include different models, portraying prototypes, assets, plans, objects paths and safety features, in this case study the ontology provides semantic representation for the necessary objects to allow inter-working between various simulation software.

The use of the concepts explained above is extensive, and each of them is used in multiple areas of development, however each concept has different characteristics that make them more effective for specific uses, for the description of objects or physical twins XML or some of its variations (AML, XAML, CAEX) are mostly used because they are easy and reliable to use; however, when the complexity of a system increases, and more effective communication among the real world and the digital world is required, it's necessary to employ ontologies, since they allow structuring and storing information that will be transfer between the physical and digital elements of the system.

3.2 Data Management

Due to the importance of the data in a DT, it is essential to carry out an appropriate data management, the main options for this are: Data Fusion and Big Data.

3.2.1 Data Fusion

The term Data Fusion includes a set of techniques, which have been widely used for data processing and data management, the use of this concept with DT technology is still arising, however, there are two sets of AI (Artificial Intelligence) techniques that can be used to perform data fusion in DT [2]:

- Probabilistic Techniques, specifically the Bayesian data fusion method used to evaluate the reliability of the obtained forecasts.
- Fuzzy Techniques, fuzzy logic is used when complications arise when modeling the behavior of certain models using mathematical paradigms (models with defined limits). Therefore, this technique is used when the limits in the models are fuzzy or when constant changes occur.

In [101] an example of the use of DTs with data fusion can be seen, the author proposes an application architecture base on a DT, for a case study of a welding production line, in this case, the data fusion techniques are used to implement the data processing, the total process contains the following stages: data acquiring, data pretreatment, data study and mining, and data fusion.

Liu [58] afirms: Data fusion processes perform a key function in the DT architecture; due to the knowledge flow, that goes from the unrefined information until smarts decisions have been made, this process is achieved through sensor-to-sensor, sensor-to-model, and model-to model fusion.

3.2.2 Big Data

In agreement with Qi [69] Big Data has been defined as the capacity to promptly obtain valuable information from a lot of data, Big Data can be considered as a technique that can be used to process data inside a DT, the combination of these two technologies allow a DT to be more reactive and forecaster, this allows a more rational and accurate management of manufacturing systems. Further, by utilizing Big Data it's possible to optimize the use of obtained data-information, because when using relevant data, decision-making and forecasts are optimized.

The use of Big Data with DT is essential when the data of a product or system evolve and reach large proportions that are difficult to manage, stored, shared, and analyzed. Although in the analyzed case studies the use of Big Data only has been necessary for one of them, which has a large amount of data to handle. Laryukhin in his work [52], develops a MAS perspective for farm managing, using the digital twin concept. The big data of the farm include: actions, effects and responses, and also parameters describing the context of the current season and the specific situation of each day (temperature, accumulated precipitation, etc.), this data is stored on a daily basis and constantly analyzed to produce a continuously updated forecast, reducing the degree of uncertainty and risks for farmers. The authors use data mining methods and tools to discover new knowledge from the stored data, resulting in clusters or patterns, that used in conjunction with ontology, allows smart decision making through the DTs.

4 Communication Trends

4.1 Middleware

According to Baken [9], Middleware also calls "plumbing", is a set of techniques created to facilitate the interaction between systems that present complex characteristics and heterogeneity. A common definition, defines a middleware as a higher level than the operating system and a lower level than the created application, it functions as an instance that abstractly represents the coded distributed system, it can be seen in Fig. 4. Thereby, makes available to programmers a set of APIs (Application Programming Interfaces) that function as a bridge between the different heterogeneous systems.

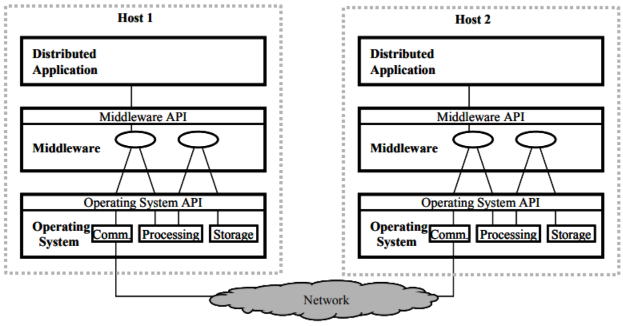

Fig. 4. Middleware Functioning

As proposed by Yun [99], every element of a DT can represent a heterogeneous system, although, for the correct functioning of the DT it may be necessary for its parts to interact with each other, whether to exchange information, negotiate or to achieve a common goal. To meet this need is possible to utilize communication middlewares, which will function as a bridge that allows interaction between the systems. Some of the most widely use communication middlewares are: DDS, HLA/RTI, MQTT, RT-CORBA. These communication middlewares are also used in case studies based on DTs, because they allow establishing a communication bridge among the real twin and the virtual twin, and also allow that the DT communicate with external systems (e.g. Human-Computer Interface) that require to extract information from the DT to properly function. Since this concept is used for the creation of DTs, is important to know the main characteristics that these middlewares present. Table 1. shows a detailed comparative among the most widely used communication middlewares.

	Communication Middleware			
	DDS	HLA/RTI	MQTT	RT CORBA
Architecture	Distributed	Centralized	Centralized	Centralized
Qos	22 policies	Reliable, Ordering	3 policies	RT
Real-Time (RT) support	Soft RT	N/A	N/A	Hard RT
Scalability	Middle	Small	Small	Small
Active Changes	Supported	N/A	Supported	Supported
Support scale	Middle	Low	Low	Middle

Table 1. Communication Middlewares

The uses of the communication middlewares with DT can be categorized as follows:

1. The **main use** of the middlewares with DT is to get information from the physical twins and communicate the DT with other systems. Schroeder in his work [82] proposes employ AutomationML to data modeling the features

of a DT (industrial valve), he also presents a high-level model whose main goal is to allow the exchange of data between heterogeneous systems, which are connected to the DT. In this case, the middleware used was FIWARE, it was in charge of extracting the data-information stored in the sensors of the physical twin and stores it in the DT. The external system is a Human-Computer Interface that shows to the users the health values (temperature, battery level, voltage sensor) of the valve. Another example can be seen in [99], where the author proposes a large-scale architecture, containing a DT-based platform to develop an ADAS (Advanced Driver Assistance System) system. To communicate the physical twin with the DT, the DDS (Data Distribution Service) communication middleware was used. Haag in his work [40], creates a DT from a bendind beam machine, the created software contains: the physical twin, the DT, the web control platform and the control module (architecture). To connect the physical twin to the DT, the MQTT (MQ Telemetry Transport) middleware was used. In this case, the middleware collects the published information on certain topics and distributes them to interested customers who subscribe to these topics. One of these clients is an IoT platform, which was used to control both twins, as well as to see the results of the realized experiments. More examples can be seen in [13, 62].

2. A **second use** is enable interactions between the DTs of the same system. Andre focuses his research work [5], on the communication layer by using DT in distributed systems, with the aim of improving the evolution of the system and making it more adaptable and configurable for different contexts, the authors create a multi-protocol communication tool (middleware), which allows the use of different communication protocols between the different elements of a holon-based manufacturing system. As a case study, the authors use the SOFAL production line, all communications between the HMI (Human Machine Interface) the SOHMS (service-oriented holonic manufacturing software) and the simulation tool (Arena Emulation) are implemented at a low level through TCP/IP connections using sockets, and the created middleware MPCT (Multi-Protocol Communication Tool) is used for communication among the real and digital twin, and also between DTs that are part of the system. Alam [3] presents a DT-based architecture reference model for a Cloud-Based Cyber-Physical Systems (C2PS). In the created system various heterogeneous systems are interconnected to achieve a common goal, each physical twin has its own DT stored in the cloud, and they are directly connected. The twins can be identified an identification number, and create communication groups with other twins to establish direct interactions. Relationships between digital twins are controlled through the Intelligent Service Layer (middleware), this layer also manages the system ontologies. Negri bases his work [62] in the creation of FMU modules, the system doesn't implement DT uniquely, the authors separate the main elements of the system into models (DTs) that will be virtualized. A middleware is also used to manage the relationships between the physical twin and the DT, and between the DTs that belong to the system.

4.2 Servers or local repositories

This section is relevant because several case studies that belong to the analyzed literature use commercial servers or communication protocols (OPC-UA), and local repositories created by the authors trough software frameworks, to obtain data-information from the physical twin and send this data to the DT, and also to communicate digital twins from the same system. Ayani [7] presents a study based on the industrial application of an emulation model (DT) to support the reconditioning of an automotive system. To achieve this, the authors propose the use of a new work methodology, they use 3D simulations through Simumatik3D Software, this software enables connectivity through industrial devices using the normalized OPC-UA (OPC-Unified Architecture) technology, the use of this technology is widely spread in the industry which makes it compatible with most PLC devices and other machines. Redelinghuys [71] performs an extension of a DT-based six-layer architecture (SLADT) using aggregation (SLADTA), this extension has the goal to create a MAS environment allowing DT-to-DT interaction. SLADT facilitates the bidirectional data-information flow among physical twin and DT, communication between the twins is done through a Siemens S7-1200 PLC and a KEPServerEX from Kepware Technologies that acts like the OPC-UA server layer, these two elements make up the local repositories. The extension realized by the authors allows establishing connections between DTs through local repositories. Others examples can be seen in [70, 100, 93, 6, 45, 48].

4.3 DT with Multi-Agent System (MAS)

According to Gorodetsky [36], a new use of the MAS technology in Industry 4.0 is arising, consists of converting the already notorious DTs into smarts agents through an agentification method. In this way, the DTs would greatly improve their capabilities, and they would have by inheritance facility to negotiate and interact with other DTs of the same system.

The analyzed case studies in which communication is based on DT technology with MAS or exists a synergy among these two technologies, use the following concepts for their conceptualization, functioning, and subsequent implementation.

4.3.1 Intelligent Product

In some case studies, the authors use MAS with DTs, this allow to provide intelligence to the DTs, according to Valckenaers [95] this creates a new concept "Intelligent Product" whereby the conduct and characteristics of the DT (Intelligent Being) and the intelligent agent can be explained and divided, the organization of the intelligent product can be viewed in Fig. 5.

1. Intelligent Beings (IB): are smart software elements who possess the ability to mirror the behavior of a product or a system during its life-cycle. Its functionality is restricted to the behavior that its physical twin can present.

2. Intelligent Agent (IA): is responsible for performing smart decision-making, for those functionalities that are outside the control of the IB, in other words, the ones that are outside the reality of the physical twin. These two classes create an intelligent product.

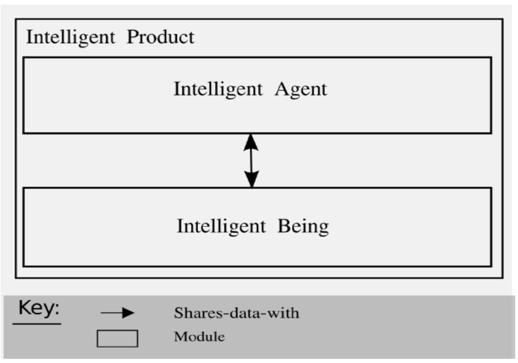

Fig. 5. Intelligent product architecture

4.3.2 ARTI (Activity-Resource-Type-Instance Reference Architecture)

ARTI (Activity-Resource-Type-Instance architecture) is a widely spread architecture, it has been used widely in some case studies to allow the synergy of Digital Twins with MAS. In agreement with Valckenaers [94] ARTI arise from PROSA [96] architecture, PROSA employ the D-MAS architectural pattern and differentiates among decision-making IA, from reality-reflection IB.

Following the work of Valckenaers, the ARTI architecture presents the following characteristics:

1. Flexible aggregation, is still a required characteristic. It represents an essential paper related to fitting into a changing atmosphere and, therefore, hierarchies vary according to the pass of time, for example reflecting the real status.

2. Autocatalytic sets (set of elements where each element can be a catalyst for another of the set) are crucial for the viability of the systems to be represented, in this architecture these sets represent the user mass of each element of the system, and they are crucial because they allow the feedback and update of information from the system, which helps to optimize its operation, also this architecture introduces a high limit for the user mass.

3. ARTI splits IB from IA, promoting the operation of the IBs. Besides splits the decision-making technologies among the decision-making processes. Thereby, ARTI works through IB, and considers decision-making as a separate process that belongs to an optional repository that can be executed within the platform.

4. Proactive behaviour, through the D-MAS architectural pattern the systems elements are able to predict the behavior of the system or the outcome result when executing a specific action.

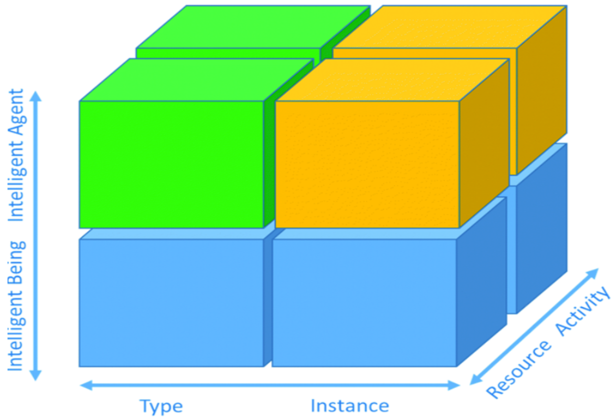

Fig. 6. The ARTI Reference Architecture

In Fig. 6. the ARTI reference architecture can be seen, ARTI evidently comes from PROSA, it substitutes the names of the entities (holons) by more general ones. Order holons are now activity instances, products holons are now activity types, and resource holons are now instance and type. As mentioned above, IBs are separated from IAs creating a top architecture feature. The IAs can be processes and instances (element in charge of communication with IBs). Staff holons remain in the green cubes; what determines if they are staff holon or not is their bond with the system.

The blue squares fulfill the flexibility theory (design considering the changes that may occur). The green squares are software technologies that can be used. The yellow squares are the set of limitations and rules that must be met (it should be limited). Is important to highlight that communication in ARTI occurs through in-depth interoperability [94].

4.3.3 Social Assets

As proposed by Bakliwal [10], 'Social Assets': are a set of assets that through mutual interactions can contribute to optimize a system. CPS are created when the set of assets are directly linked to software elements (computers, controllers), which the ability to execute algorithms and make decisions on the assets. Every asset has its corresponding DT (smart agent), which reflects its behavior in real time and makes decisions using the data-information that comes from the asset. Through MAS a set of DTs create a network named as "asset fleet" [26], by forming networks the DTs become more reactive and gain the ability to obtain information about events that haven't affected them yet, which allows them to have a better knowledge of the environment that surrounds them, and a broader dataset to perform training, this also improves the forecast capacities.

The concept defined in the previous paragraph is an emerging concept, and its main contribution lies in communication, because the elements that are part of a network of industrial assets act as social entities and are capable of sharing information and collaborating with each other to perform a specific task, or for optimizing the system.

In the realized research, this concept is used in two case studies, Palau [66] uses it to create networks of industrial assets, in which the assets share information, which allows identifying and sharing failures (collaborative learning), and data sharing for improve prediction accuracy (collaborative prognostics). Kumar [51] also creates industrial asset networks, in which assets exchange information to deal with the NP-hard problem of Job Scheduling, the interaction between these assets (industrial machines) lies in sharing data, or computational results to optimize the schedule created by the system.

4.3.4 Model Type Identification

Jung in his work [44], creates a new concept 'Model Type Identification': it's a general method to group the models (IoT simulated devices) according to the communication protocol they employ. For example, the models will be grouped into a "Bluetooth" or "Wi-Fi" group depends on the utilized communication protocol.

Following Jung's work, the author uses the created concept to optimize communication between the agents of a MAS, in his approach he creates a MAS for dynamic co-simulation, each model that belongs to the DT of a system agent can be simulated in different simulation tools, and each simulation tool has a different communication protocol, using the concept 'model type identification' allows agents to be grouped according to the protocol communication that his model use, and create a communication interface for each protocol, thus standardizing communication between agents and reducing message traffic, since sending messages is only allowed between agents belonging to the same group.

4.3.5 Case Studies

The following 10 case studies are the most meaningful ones regarding the use of DT with MAS in the realized research.

4.3.5.1 Case Studies with ARTI

The next case studies employ the ARTI architecture concept for their functioning. Cardin in his work [20] introduces a generic DT-based framework with the particularity of measuring the energy used by the modeled manufacturing system. This framework works by coupling two main models (multi-physical and behavioral) to create a real-time simulation to mirror the main system behavior, and to realize anticipated predictions to improve decision-making, the authors use the ARTI architecture to integrate the models into the digital twin, the authors utilize ARTI because this architecture is the ideal one, since it's better adapted to manufacturing systems and natively integrates the DT concept. An energy-conscious digital twin is created using several blue cubes through the IBs. The energy awareness works through the measure of the used energy by the system resources. This measure is realizing taking into account physical processes and characteristics of the environment, therefore, the coupling of the following models is necessary: (a) multi-physical, describes the physics of the system and returns approximate estimates of energy measurements based on information from the system; (b) behavioral model describes the logic of the system and its relationships with the environment.

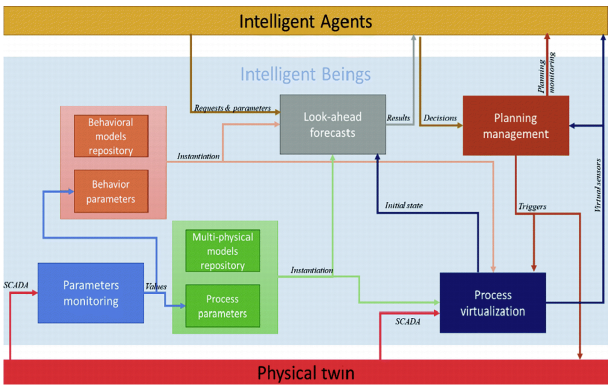

Fig. 7. Energy-aware resource architecture for DT

In Fig. 7. the behavior of the created framework can be seen, it's also possible to observe that there are six main functions: behavioral models repository, multi-physical models repository, parameters tuning and monitoring, process virtualization, look-ahead forecast and planning management, the red bottom level represents the relation with real devices of the system.

There is an intelligent being (DT) for each of the elements that are part of the system, inside the IB the coupling of the multi-physical and behavioral models is executed, through these models the virtualization process is performed. In the look-ahead method, the virtualization process method is employed to supply the required data-information to initialize the forecast, this method employ the DES (discrete-event simulation) simulation through the Rockwell Arena software.

There is an Intelligent Agent (MAS-Agent) for each of the elements that are part of the system, the IB is in charge of exploring possible decisions using the look-ahead forecast function, the results of this function are sent to the IA who makes the corresponding decisions using their intelligence. The decisions made by the IA are sent to the planning management function, through this function the IA communicates with the physical twin and from the decisions made by the IA the planning execution is triggered in the physical twin. It's important to highlight that there is communication between the IB and IA of each element of the system through the planning management function and the ARTI architecture.

The authors applied the proposed methodology to a thermoplastic injection process to measure energy consumption to check the correct operation of the framework, the process consisted of the following steps:

1. Create a multi-physical model based on the injection process, which will have the ability to evaluate energy profiles over periods of time and coupling it with the behavioral model.
2. Perform energy measurements in the real system under specific conditions.
3. Adjustment of the multi-physical model taking into account the information obtained in the previous step.
4. Create a DES model of the real system.
5. Adjustment the DES model with the data-information received in real time.
6. Once the DES model is adjusted it can be used to make forecasts by the IB, and for decision-making and management planning by the IA.

To analyze energy consumption, the authors divided the machine into elemental consumption equipment to identify the corresponding models. The idea behind this is to determine for every existing resource, an energy profile related to every specific action. After performing the energy measurement for 40 injected products, the authors conclude that the implementation was successful. This article has great importance since its main objective when measuring energy consumption, is to optimize production management by measuring key performance indicators (energy), which presents great advantages for Industry 4.0, since it would reduce costs and processing time.

Borangiu [15] creates a framework based on an embedded aggregate DT for the hybrid management of a semi-continuous production process of radiopharmaceuticals. For this, different twin aggregations are defined: (a) DT of the three main threads in the production plant; (b) predictive twins and twins in charge of decision-making and their projections, at last, the DTs are integrated into the ARTI architecture with the objective of optimizing the production process. As a case study, the authors use a radiopharmaceutical production system that is based on semi-continuous manufacturing processes. The radiopharmaceutical system works on a cyclotron basis, the system groups the given orders in the past 24 hours in potentially production intervals. Although the system has a specific structure, each solicited item walk through the next manufacturing process: (1) radioisotopes produced in a particle accelerator (cyclotron), after (2) they are moved to technological isolators to suffer chemical synthesis, later (3) distributing: at some point, dilute and separate the majority of the product into vials, ultimately (4) quality control of samples of finish products through multiple performance tests. The processes 1 and 2 are constant, while the 3 and 4 are managed by discrete algorithms.

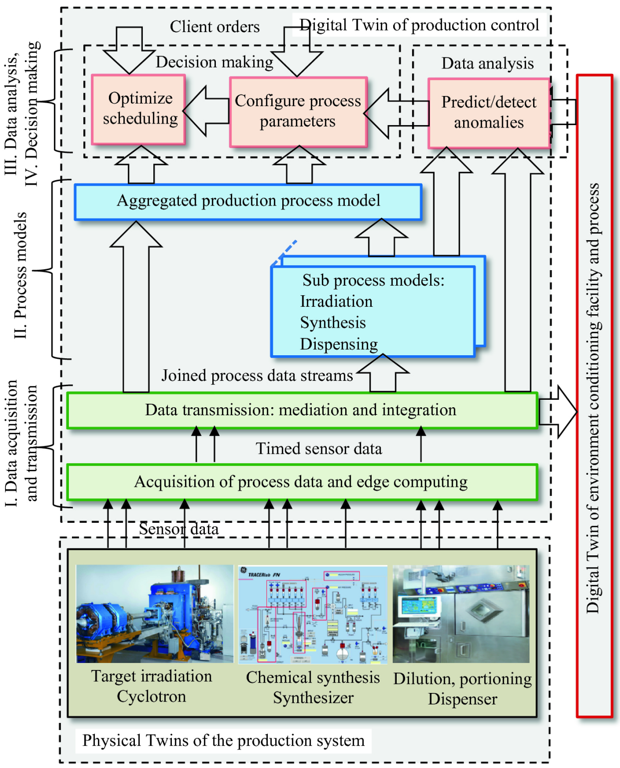

Fig. 8. Aggregated DT embedded methodology

To achieve the optimization of the system, the authors use the following DT methodology (Fig. 8.):

1. Data acquisition and transmission: the DT includes the most relevant data-information and pass this information on to the next steps.

2. Virtual twins of sub-processes: a DT is created for each of the main sub-processes of the system (irradiation, synthesis and distribution). The creation of these instances allows abstract and facilitate the interaction between the physical world and the virtual world.

3. Data analysis: this layer is composed by several predictive twins, which have the ability to acquire data-information to realize training sessions (neural networks), and through the results of these sessions they can perform accurate forecasts and identify failures in the system.

4. Decision-making: algorithms that receive the forecasts from the previous step, and use this information to take decisions to optimize the system, they play a double role: manager (taking decisions) and actuator (realize corrections on the system).

The DT acquires data through interactions with system sensors, through IoT gadgets (preprocess the information); these gadgets are linked with the virtual twins of the three main sub-processes.

Furthermore, the authors apply through the ARTI architecture the holonic paradigm to hybrid supervised control (H2SC) of semi continuous production systems, in this case using a radio pharmaceutical case study. The implementation of holonic theory has the following steps:

1. Define the most important parts of the system to assign a DT to each of them, define how they will be connected and the models that this connection need to properly functioning.

2. Define the holarchy and the elements that will become holons to apply the H2SC; to achieve this goal, the authors employ the ARTI architecture and a MAS framework.

Using the predictive twins it's possible to detect failures and malfunctions in the system behavior, and using the twin's algorithms it's possible to infer possible solutions for the problems that come up. These twins are inside the ARTI architecture specifically as "intelligent agents".

It's important to note that the digital twins that represent the components of the manufacturing system, communicate with each other through the FIPA ACL Standard communication protocol, and trough the agent in charge of managing the system.

This article present high relevancy because it acts as a topology management in which initial parameters are established, these parameters optimize the cost methods, if the configuration remains the same during the execution there isn't process alterations. However, if the integrated virtual twins detect variations in the setup or failures in the system, the intelligent supervisor agent creates new actions instances (if possible), allowing the system to operate autonomously by identifying and correcting anomalies.

4.3.5.2 Case studies with Social Assets

The upcoming case studies utilizes the Social Assets concept for their perform. Palau in his work [66], creates a MAS architecture framework which allow cooperative learning and real-time distributed cooperative forecast. The system is formed for social industrial assets, these assets collaborate and exchange information between them to generate optimal enterprise level, this collaboration is allowed by calculation inter-assets similarity during operation time to recognize 'friends'.

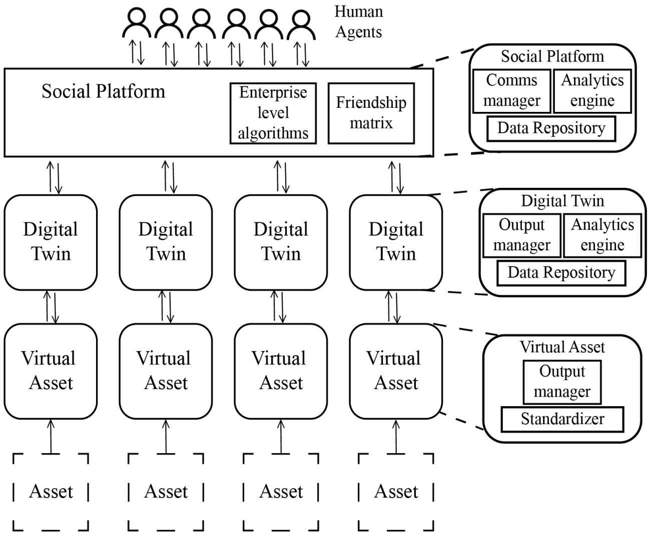

Fig. 9. MAS architecture

The created architecture can be viewed in Fig. 9. and is formed by the next elements:

1. Virtual Asset, is a software component, exists one for every asset, which main objective is to normalize the heterogeneous data sent by the asset before it reaches the DT. This software receives the data-information through a file sent by the asset, and realizes a process to normalize it, finally send this information to the DT. The data-information is formed by the following elements: a group of characteristics (coming from the sensors), a group of temporary events (when a malfunction occur), and an asset identification number. The Virtual Assets are composed by two building blocks: a Standardizer, dedicated to standardizing the data coming from the Assets, and an Output manager, that controls the communications with their assigned Digital Twin.

2. DT (smart agent), every virtual asset has his own DT, the DT is formed by three main elements: a Data Repository, an Output Manager, and an Analytics Engine. The Analytics Engine (set of algorithms) performs the following actions: forecast, data-information

normalizing, analyzes data-information received from the Virtual Asset and the Social Platform, which allows it to classify the received events in: related to forecasts or cooperative learning and malfunctions. The Output Manager manages everything related to interactions either with Virtual Asset or with the Social Platform. Lastly, the Data Repository is where all the data that comes from the DT is deposited and managed.

3. Social Platform, is the element in charge of managing the interactions among the DTs, and communicating the orders received by the users to the system. Utilizes enterprise-level algorithms to: (1) creates groups of cooperative assets, and (2) manage enterprise data-information, to create the groups execute a clustering algorithm as a parallel thread. The friendship matrix stores the identification number of each asset and the distance (euclidean distances) among the assets, it's also used to measure the weight of cooperation among assets. The Social Platform is composed by the following elements: a Data Repository, storing the Friendship Matrix and the outcomes of the enterprise-level algorithms, a Communication Manager, that manages the system interactions, and an Analytics Engine that executes the algorithms.

The authors highlight that in an asset fleet a disastrous error might happen to a specific group of assets, therefore collaborative learning is profitable since it allows the assets that are communicated through the Social Platform, to exchange data and knowledge. For the cooperation between the assets to be productive, is essential that the data information being shared cover all the relevant aspects of the system.

Collaborative prognostics, consists of grouping assets with similar characteristics and allowing these assets to interact with each other to share information and knowledge, allowing them to make a more accurate forecast. To realize real-time forecast the authors decide to utilize the machine learning approach WTTE-RNN (Weibull Time To Event-Recurrent Neural Networks).

As a case study to prove the architecture functioning the authors employ the Turbofan Engine Degradation Simulation Data Set (C-MAPPS), C-MAPPS is a MATLAB-based software created to emulate the behavior of a real turbofan engine [78]. The software implementation is coding in Python, specifically Keras, Tensorflow [1] and use the wtte-rnn Python package. The system employ the websocket library to allow interactions among the DTs, and the asyncio library and threads to guarantee asynchronism and parallelism. For clustering, a DBSCAN (Density-Based Clustering) algorithm was used. Finally, the authors conclude that the methodology created and tested with the case study presented shows the benefits of using Social Assets and collaborative prognostics.

This work is pertinent to the research because it uses the 'Social Assets' concept, to integrate interaction abilities present in today's society, such as social networks in Industry 4.0, these abilities are the ones called SIoT (Social Internet of Things), by using this concept the authors creates a system that allow not only form groups (clusters) of agents that share similarities (distances) and data exchange between these agents, but also makes use of this relevant information so that the agents learn from the failures of their 'friends' (collaborative learning) and for improve their prediction accuracy (collaborative prognostics). Authors are able to realize this approach by creating an organized level hierarchy which in each level presents different analysis algorithms.

Kumar [51] proposes a Multi-Agent System based distributed operations planning approach for scheduling jobs in a parallel machine shop-floor. In the created system each asset is represented by a smart agent (digital twin), agents interact with one another to form a network of social machines. Using distributed decision-making and communications within the network by using the 'Social Assets', concept the authors deal with the NP-hard problem of job scheduling.

The authors considered a job shop having parallel machines. And they assume that the machines are made up of single component. These machines are identical in the sense that the jobs in demand can be processed on any of the machines. In addition to that, the time taken for production of the jobs is same irrespective of the machine which processes the job. However, the machines can be different from one another in terms of their reliability, and their age. The reliability of the machines is characterized using the two-parameter Weibull probability distribution. These Weibull parameters are computed, using the historical data pertaining to the failure of the asset, by their corresponding agents. They also assume that there is only one mode of failure for the machines and hence a set of Weibull parameters completely describe a machine's operating condition.

In this scenario the Production Planning and Control (PPC) department receives the job demand (job names, job descriptions, processing and due times, penalty) and is responsible for creating final job schedule for all the machines in the industry. The PPC and the machine agents are two levels which collaborate over their asset data to coordinate and fulfill the task of producing the enterprise-level job schedule. The authors also considered the Preventive Maintenance (PM) tasks, it's important to schedule PM-Jobs in an optimal way cause it reduces the machine age, preventing failures.

The authors create a few assumptions: a) A job can't be outdone by another ; b) At the starting point all the jobs are accessible; At the starting point all the machine are accessible; d) Machines can only handle one job at each occasion e) Machines produce adequate quality products every time. With the goal of create an enterprise-level job-schedule which minimizes the penalty cost, in this case the penalty cost is the same for all the jobs, so is necessary minimizing the lateness of production of the jobs.

All assets have its own embedded micro-controller or a processor, which allow agents control. The asset data reflects its working condition, on which the local computations are based. Thus, the corresponding agent is capable of independent decision-making based on the asset's data. An agent further linked to the agents of other assets to enable human-like interaction. Interaction here means the sharing of data, or the computational results at different levels.

The created approach can be viewed in Fig. 10, and present the following methodology:

1. The job demand reaches the PPC department.
2. PPC Department circulates the job demand across all machine agents.
3. Machine agent analyzes their respective past time-to-failures data and fits a reliability model (Weibull distribution) to represent the health of each machine.
4. The machine agent uses heuristics to generate a set of optimal job sequences (minimum cumulative lateness), if the health goes above a certain limit, it schedules a PM-Job, according to the required objective.
5. Each job-sequence generated in the above step is assigned an index of feasibility for comparing with other sequences, generated by the same machine or other machines, at the enterprise-level.
6. List of job-sequences from each machine is sent back to the PPC department which evaluates them and prepares the enterprise-level job schedule.
7. PPC communicates these final schedules to all the machines for execution.

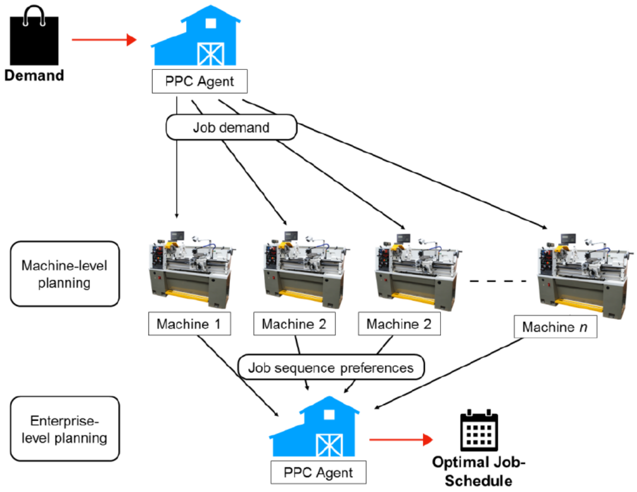

Fig. 10. Distributed MAS Approach

The created methodology was evaluated against the centralized approach (done at single level) in 9 cases. Job characteristics and machines properties are programming in Witness 14 simulation framework for all cases. For optimization, Adaptive Thermo-statistical Simulated Annealing (ATSA) and brute force (evaluate all combinations) have been used, the brute force was depreciated because of time limitation. The authors conclude that the distributed ATSA approach shows significantly lesser computation time that the centralized ATSA, thus, distributed approach can be considered as a promising approach to solve complex shop floor scheduling problem.

This article is innovative because, as the authors point out it's the first work that presents a heterarchical multi-agent system with distributed decision-making, which involves coordination among multiple agents (smart digital twins) of the system through the 'Social Assets' concept, i.e. the machines and the PPC division, to deal with NP-hard problem of job scheduling.

4.3.5.3 Other Case Studies

Alaya [4] presents a self automated quality management framework supported by CPS and MAS. The proposed platform is validated through a case study.

Propose paradigm can be viewed in Fig. 11, and presents the next components:

- Sensing module: This module represents the CPS-agent entrance point of the incoming data. CPS network is based on an IoT infrastructure, that means that every CPS-Agent is equipped with a network interface, this interface is the sensor responsible for collecting logical data, which is mainly messages from others CPS-Agents and commands from the information system. Physical sensors are part of the sensing module responsible for collecting physical data. Plug and play functionality is required to enable self-configuration.

- Pre-processing module: This module acts as a physical filter, it's responsible for row data cleansing. It's also responsible to assess the data accuracy based on the level of confidence assigned to the source. In the case that several sensors are implemented, the data fusion is performed in this step in order remove incoherence. The output of this module is the pre-processed data used by the decision module.

- Decision module: The added value of a CPS resides on its decision capacities. For this reason, this module is considered as the most important. It has the mission to process the received pre-processed data to provide decision. The cognitive capacities used are based on Artificial Networks and Fuzzy Logic to perform essentially prediction tasks. The predictions made are used to automatically self-configure its behavior, optimize the production incomes, reduce impacts and avoid failures.

- Validation module: The validation module is responsible for sending commands to the actuators. This module is essential to check the validation of the decision proposed by the decision module. On the other hand, this module checks if the proposed decision is compatible with other ongoing process and does not create conflicts. To do that it uses optimization methods like scheduling algorithms.

- Actuators: These are components in charge of executing the received commands. They can be network gateways sending alerts, reports or messages to the concerned parts. They can also, as commonly known, seen as entities able to act on the physical environment.

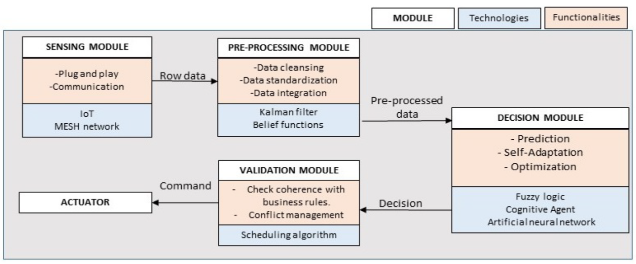

Fig. 11. Proposed Framework architecture

The proposed framework is implemented in a quality control system, an industrial partner APR (Application Plastique du Rhone) which is a French company specialized in plastic manufacturing based on their criteria. The proposed approach is executed through a Raspberry Pi card, using JAVA language; also utilizes: an HD camera with a mobile arm, a LED table providing white light to generate negative image for patches detection, and a 360 Degree Laser Range Finder (Lidar) to precisely detect the items position. The CPS-Agent is constantly soliciting the information system via web-services requests.

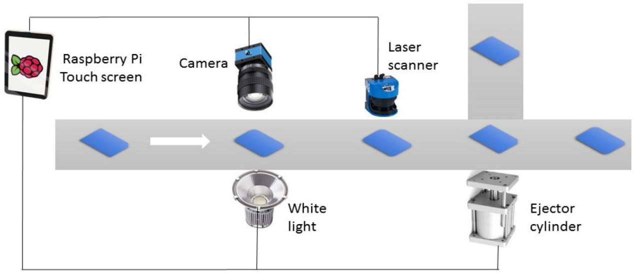

Fig. 12. Conveying plan proposal for quality control

The solution that adopts the architecture is described by the Fig. 12. The physical entry of the system is the item to control. The logical entry is represented by the configuration files of available sensors and actuators. A measure of the item is first sent as input of a web request to the information, trough this the CPS agent is able to sort the manufactured items into compliant or faulty which is the main functionality of the system. It's also able to predict problems, raise alert, calculate impacts and adapt its behaviour based on its observations. This article is relevant because it deals with one of the main problems produced for the product customization trend (quality control), creating a feasible, scalable, autonomous, and plug and play hardware software solution which provides advantages for the Industry 4.0

Jung [44] presents a new concept for dynamic co-simulation for IoT-systems using a muti-agent system. In this system, each IoT-element has a digital twin (smart agent) and is simulated in a separate simulation tool, the DT is able to enter a running co-simulation dynamically during execution time, allowing the concept of 'Plug-and-Simulate'. The communication among the agents and simulation tools is carried out by an interface.

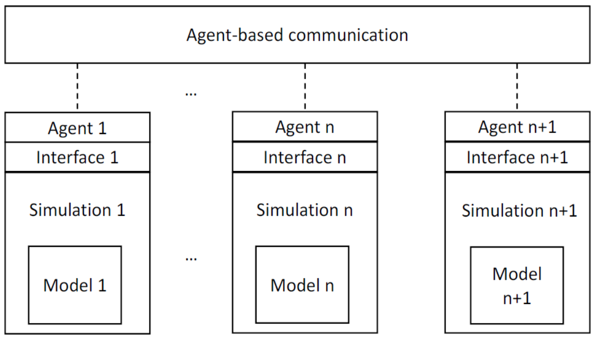

Fig. 13. Multi-agent-based co-simulation

The authors underline that MAS was selected for coupling the simulation tools since the smart agents are capable to entry or leave during run-time execution. As mentioned in the previous paragraph each IoT-element is simulated in an individual simulation, probably with a different simulation tool, each individual simulation is portrayed by an agent, this can be seen in Fig. 13. Through simulating each individual component and this being represented by an agent, the ability to swap models at runtime is acquired. The interaction between the models is controlled by each agent.

Because the system is formed by IoT-elements, the models have the capacity to interact with each other both virtually and physically the authors control this interaction in a process-oriented interface. Also, a framework to manage the synchronization between the models is necessary, because they employ different simulation tools, this framework will communicate direct with the agents. The interactions with the agents work for each one equally, however the connections of the agents with their respective models present particularities derived from simulation tools. Each process of the interface has two parts, a generic part that is commonly used (connected with the agents), and a specific part for particular use (connected with the simulation tool). The specific part must be created for each simulation tool separately, all the interfaces can be viewed in the Fig. 14.

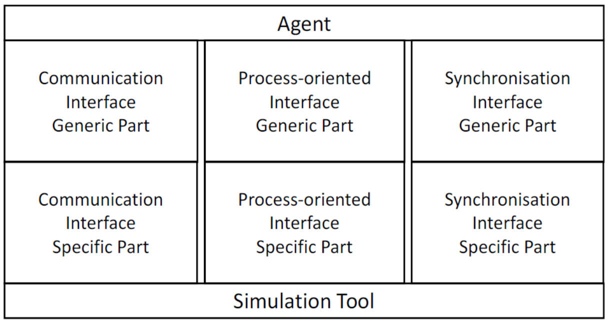

Fig. 14. Agent-simulation-tool-interface

Based on the 'model type identification' concept the models of the system will send notifications only to the models that are part of their group. The authors also create an interface for each communication protocol (Bluethooth and Wifi).

To prove the created concept a smart warehouse scenario was utilized, this scenario has the following assets: a storage rack, a set of sensors and a forklift, these assets can communicate to each other to optimize the functioning of the warehouse (storage). The storage rack have an embedded sensor which determines whether an asset can be stored or not (depending on the temperature). Every time an asset arrives at the warehouse, it requests to be stored sending a request, once the request is received by the storage rack check its current temperature to make a decision and sends an answer, if the answer is yes the forklift collects the asset and places it in the storage rack, and finally sends a notification to the warehouse notifying that the asset was successfully storage.

The simulation software used to implement the scenario were MATLAB Simulink and OpenModelica. Jadex (agent architecture) was used because it allows the dynamical co-simulation among the created models. A graphic user interface was created to allow users to choose which models are going to be plugged or unplugged. The authors conclude that the created concept allows that the multiple models to be connected or disconnected depending on of the users request.

This paper presents a high value because it deals with the communication problem that heterogeneity causes in an IoT system, and also allow to prove the Plug-and-Simulate concept. The authors achieve this goal, by creating a system in which each element (IoT model) of a system is modelling in different simulation tools, the communications is manage by a super-ordinate simulation among the elements, rather of separately translating the interactions among each simulation tool. This super-ordinate simulation works inside a MAS, this system also allow the Plug-and-Simulation concepts using agents ability to enter and leaving the system.

Laryukhin [52] develops a multi-agent approach based on CPS and DT technologies, for managing precise farms. The approach is knowledge driven, digital twins of plants are created, each DT is a smart agent created from a set of relevant information to optimize plant cultivation (plant type, crop, etc), also contains historical data about environmental conditions and activities.

The main idea of the proposed approach is to consider crop cultivation as a complex adaptive system with distributed collective decision making among crop varieties, soil and fertilizers, precise machines, etc. [77].

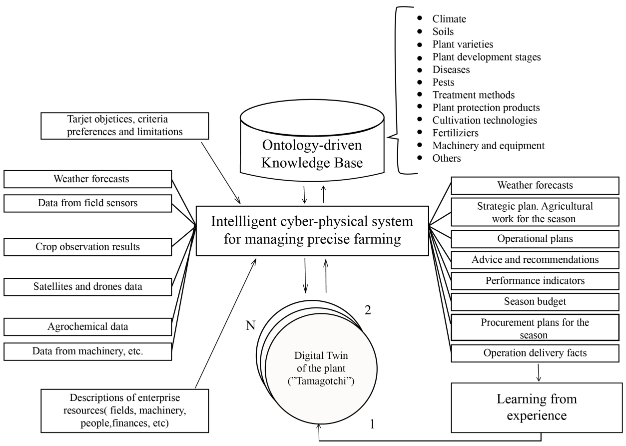

Fig. 15. CP MAS for precision agriculture

The developed Cyber-Physical Multi-agent System (CP MAS) for farm management integrates sensors, actuators and communication tools, and, on the other hand, computer models of plant growth and development combined with knowledge base and decision-making tools for farmers, it can be viewed in Fig. 15. The authors based their investigation in the hypothesis of that the reasoning of agronomist and other specialists in farm management could be modelled as a process of above entities self-organization which can be implemented with the use of multi-agent technology.

In the created system agents form demand-resource

networks (DRN) and operate on an internal virtual market with the opportunity to buy and sell time slots in the schedule. During this process agents are able to discover problems and solve conflicts applying negotiation protocols and compensation method based on individual satisfaction functions and bonus-penalties. The result of the interaction of agents inside the CP MAS, is the plan for crop cultivation, which can change to adapt to the occurred events.
The followings agents were created:

1. Agent of farm, to maximize objectives given by farm owner.

2. Agent of plant, to maximize plant productivity and efficiency.

3. Agent of soil, to maximize richness of micro-elements and water in soil.

4. Agent of fertilizer, to maximize quality of crops, etc.

The authors use ontological models based on the agronomists experiences to create the base knowledge, the models are based in the following topics: climate, soils, plant varieties, plant development stages, diseases, pesticides, treatment methods, insects, plant protection products, cultivation technologies, fertilizers, machinery and equipment. They also use the big data concept, with data mining techniques to create clusters for decision-making support.

The Digital Twin of a plant is created for each field to mirror real plant growth and development. It would mirror plant development on everyday basis, representing the most predicted version of the plant development plan, updated daily with data from the weather server, sensors in fields, observations of agronomists, etc. So using the CP MAS the system allow play "What if" games and form the most perspective scenarios for the season, forecasting yields and risks for business. The main components of the system digital include digital IoT platform which provide data and commands to farmworkers, CP-MAS for decision-making support, digital twin of plant, simulation, forecasting modules, pattern recognition of situations, ontology editor and editor of plant model which provide farm planning. The architecture of the CP MAS can be viewed in Fig. 16.

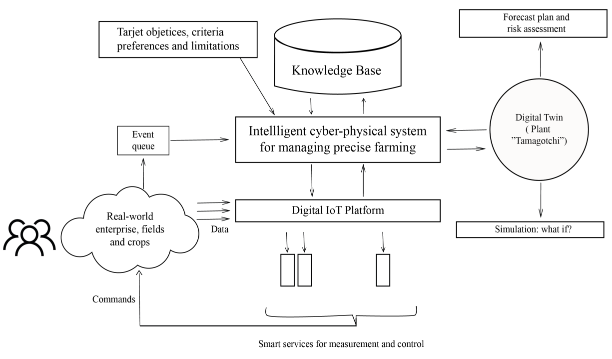

Fig. 16. CP MAS architecture

This work presents high precedence because it deals with the problems of using precise farming technology, these problems are complexity of technology and priory incompleteness of knowledge about factors of plant growth and development, cause each crop, each farm and each season is unique, besides using precise farming requires a high knowledge about biological, chemical and physical processes in plant and soils, in addition is necessary to taking into account the climatic changes that may occur. To solve this problem the authors create a CP MAS for farm management, this system is knowledge driven, it's specifically based on the knowledge of agronomists, characteristics of crops, crop history and environmental conditions. The system use this knowledge to create different intelligent digital twins that through negotiations give as result the crop cultivation plan. Using this system will upgrade the quality of the produced product and efficiency of crop cultivation inside the agriculture industry.

Gorodetsky [35] proposed a conceptual framework to develop an autonomous CPS for adaptive resource management (ARM). When cyber-physical multi-agent systems (CP MAS) are used for ARM, the DT data is modeled using ontologies and real-times simulations are allowed, to improve the resources scheduling.

The authors use a basic ontology to solve the resource management issue, that models the main concepts of the system as classes and describes the relationships between them, which allows condensing the most relevant information related to resource control in the company. These classes are associated with intelligent agents which create "instances" of them, test their operation and carry out simulations with specific characteristics.

Using the basic ontology, an ontological industry model is created when instantiating the concepts (classes) and relationships. The created ontology is based on the company domain, and by using the ontological industry model the ontology driven DT (ODDT) is created. The ODDT reflects the life cycle of the company, if there is any change it will be updated, it also realizes future predictions to optimize the system. The conceptual framework can be seen in the Fig. 17.

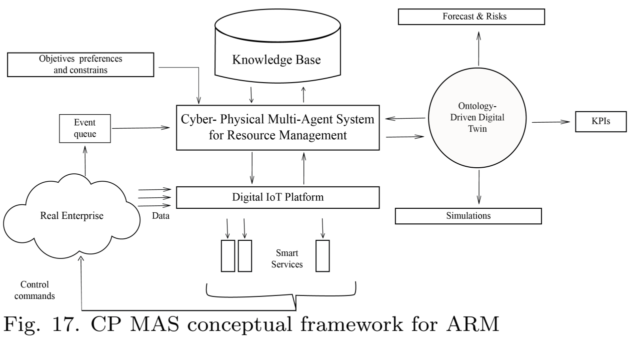

Fig. 17. CP MAS conceptual framework for ARM

The ODTT agents are reactive, and have the necessary intelligence to: take crucial decisions, negotiate the resources management with other agents or users always looking to maximize their profit; also support real-time simulations and what-if games; besides can realize previsions to detect failures and evaluate decisions contrasting key performance indicators (KPIs), especially in relation to resource management. The system users have full management of it, being able to modify the objectives and can also add or delete information from the ODDT.

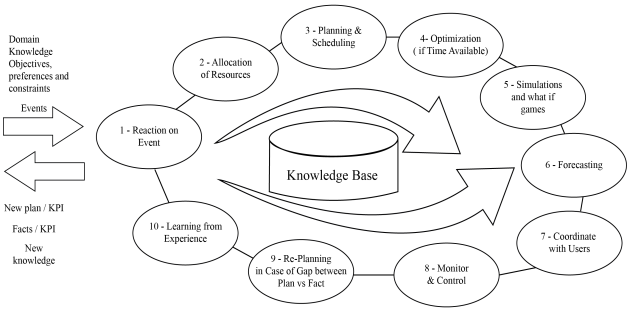

Fig. 18. Life cycle of autonomous CP MAS for ARM

The created framework main objective is to control the resources management during the entire life cycle of the system. The system life-cycle can be seen in Fig. 18, and is mentioned by the authors as an extension of the Deming cycle 'Plan-Do-Analyze' [25].

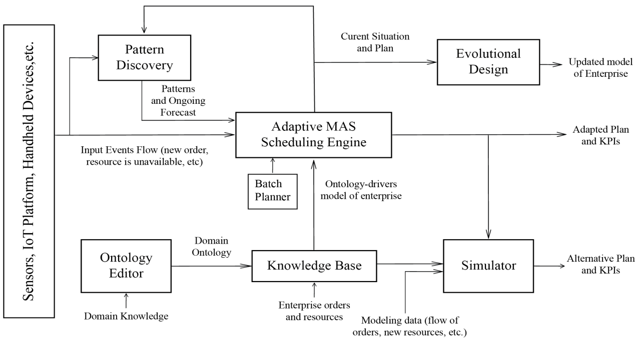

Fig. 19. Architecture of CP MAS for ARM

The principal components of the developed framework are (Fig. 19):

- Ontology Editor.
- Knowledge Base of the company, include the principal concepts of the system and the relationships among them, is essential to develop the ontological industrial model.
- Action Queue, in charge of storing received actions sent by the users, has a set of rules for these actions.
- Adaptive MAS Scheduling Engine (set of algorithms), in charge of taking smart decisions, and conducting negotiations between agents, it also contains the negotiation protocols and the bonus-penalty methods.
- Batch Planner, contains the basics tool-kits to improve the initial planning schedule.
- Pattern Discovery, clustering tool and data mining.
- Evolutional Design, tool in charge of reconfiguring the system if a failure occurs.
- Simulator, can copy scene from Adaptive MAS Scheduling Engine and run simulations in real time for forecasting, can be executed in parallel.

The principal output of the created framework are: optimized scheduling plan, optimized KPIs; besides the system have the ability to evaluate the decision-making and the behavior of the system when an event o action occurs.

This paper contributes to the manufacturing area allowing ARM, cause the use of the created system allow systems to have the ability to optimize the resources management during the entire life-cycle of the system; through simulations executed in parallel which explore the possible paths that the system can take. This allows the system to be reactive and work more efficiently.

Rodemann [75] develops a MAS framework, which consists of creating a model of the behavior of a set of clients to evaluate EV (electric vehicle) charging systems, specifically to determine how customers behave when it gets involved outside conditions (environmental) or past experiences (previous interactions with the system), and how client behavior influences system performance.

The charging system is simulated using (SimulationX [28])based on Modelica [29] standard. The charging system interacts with two heterogeneous systems through an FMI, the charging control unit is coding in Python and the environmental conditions is simulated by NetLOGO. The charging control unit contains all the necessary information to manage the loading processes (through a set of algorithms), to be able to realize this management receives the following data: characteristics and usage information of the charging stations, and the vehicle parameters upon arrival.

The created MAS framework includes three distinct agent kinds and a simulated environment. Each agent has its own features, events, and functions, several instances of the same agent can coexist. In the proposed case study, exists the following agents; 30 'e-vehicle', 21 'charging point', and 30 'driver'. The simulation environment also includes external conditions (weather).

- Agent Driver, represents a real person driving a car and has a daily routine, his goal is going from point A to B with enough battery charge in his vehicle. Decides when to charge or not his car through a particular limit (individual threshold value). This limit depends on the vehicle location (company o external charging infrastructure).

- Agent E-vehicle (EV), is the virtual copy of a real vehicle, its parameters are included in the simulation model. Has the following features: charging structure and status, energy intake, and the basic status of the vehicle (driving, parking, loading); receives ambient conditions data through the FMI.

- Agent Charging Point (CP), every charging station has its own charging agent, the agent represents the station in the charging station system (CS), presents particulars charging features. These features allow the 'driver' to charge the 'e-vehicle, only if the features are equal with the ones in EV agent. It has the ability to store and export its information.

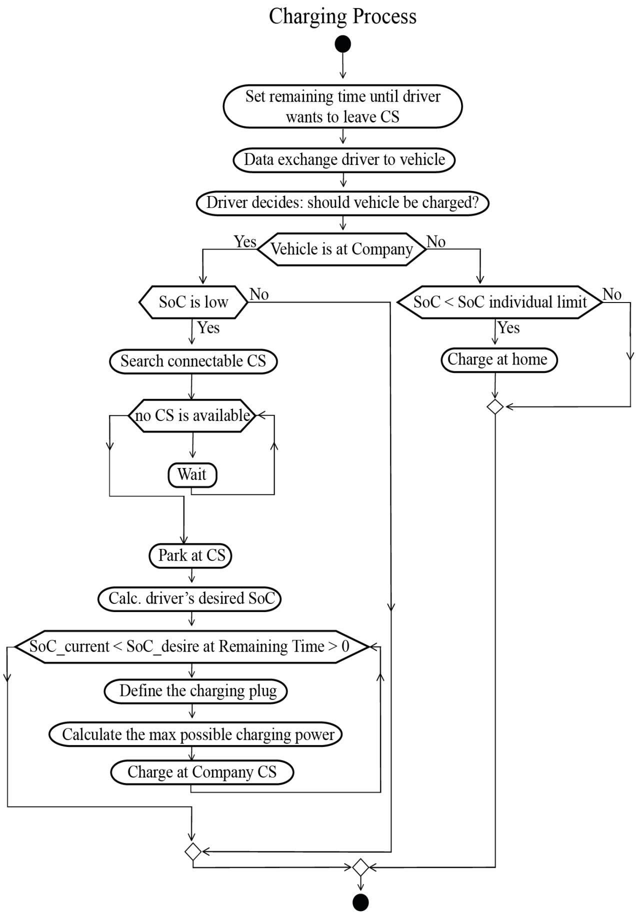

Fig. 20. Modelling of the charging process in MAS

The authors created two distinct situations, representing different behaviors from the customers. The first one is a basic expansion from the basic situation, where the energy request increases depending on the weather conditions (hot or cold). The second one represents a situation where the client decides to stop using the system if he had bad previous experiences (customer satisfaction modeling), he takes this decision based on the desired state of charge (SoC) when arriving at the system, and the SoC obtained at the end of the interaction with the system. The modeling system was a medium-sized photovoltaic (PV) company with a peak power of 490kw. The CP hand the EVs have a maximum load of 10 kW and 150 kW.

The charging control unit calculates the charging power available in the system and the extra energy that comes from the network (PBase) in each charging process. In order to minimize the appearance of energy peaks, lower expenses, and minimize the *CO2* emissions, PBase has to decrease its value. The customer satisfaction indicator (CSI) was calculated using a single function of each charging process. The functioning of the created approach can be viewed in Fig. 20.

The authors simulated the two defined scenarios for 365 days using the weather conditions from a city in Germany. The authors conclude that both situations have at outcomes important differences related to system efficiency (used energy, charging periods). The situation where customer behavior was modeled based on historical data had better results, the unsuccessful loading processes were almost non-existent. It's also possible to infer that increasing the available energy had positive results, however, it wasn't able to maximize customer satisfaction.

This work is innovative because it deals with the problem of developing an efficient electric vehicle (EV) charging infrastructure, modeling customer behavior allows has a better understanding of the system and make decisions to optimize it. In addition to using other elements to understand customer behavior, it allows leave the classic stochastic system where customers were modeled passively.

Clark [23] creates a language expansion which includes agents with learning and operations capabilities, the authors also present its performance (ESL platform developed by the authors), describe its mains characteristics, and present a case study base on a supply chain system.

The authors hypothesis the use of Multi-agent reinforcement learning (MARL [18]) to deal with the inherent uncertainty in systems that are going to be executed in an environment where state changes are constant. For use MARL there has to be a mechanism by which collaborating agents, each with their own learning capacities could be combined in an optimized manner (i.e, reinforcement learning), that's why the authors create a language extension support for MARL.

The ESL platform for MARL development use actors to create digital twins of complex systems. By adding goals and intentionality to actors it arrives at intentional, autonomous, composable modular units called agents. Each agent try to achieve it stated goal by responding suitably to the events and by communicating with other agents, however, each agent must be reactive and modify their behavior according to the environment conditions, and to interactions with other agents, that's why Reinforcement Learning (RL) is a promising approach that can easily be integrated with agent behavior since it supports the incremental improvement of selecting local actions based on an agent's observation of a global environment.

The created framework presents a collection of agents each with a message queue, a state and a behavior. The behavior is envisaged as a non-deterministic state machine whose transitions are controlled by a Q-table [22]. Agents communicate by sending messages which are queued when they are received. Each message is processed in order in each system cycle. An agent has the following type *Agent[S,M,A]* and consists of a state *s::S*, a collection of messages *m::M*, and a collection of actions *a::A*. The goal of the agent is specified by its terminal state: it stops when it reaches goal. The reward is calculated in each state. If the state is currently reaches 'goal' the rewards is the length of the agent history, then the agent is required to minimize the reward over the history, reinforcement learning will tend to favour shorter histories that lead to goal.

In order to ensure an agent achieves its goal, it must be trained. Training involves running an agent for several *epochs* using the reward function to evaluate the history of each run. Once trained, an agent can be run using an ESL function. ESL also provides a collection of system building operations so that composite agents can be constructed (agent of agents), trained and then decomposed. A constructed agent support inheritance, each agent may extend a parent and will inherit his father's learning. Also, once an agent has been trained, it may be desirable to reuse the learning in a different context, or to compose learning from multiple agents by mapping their Q-tables to achieve a common goal [84].

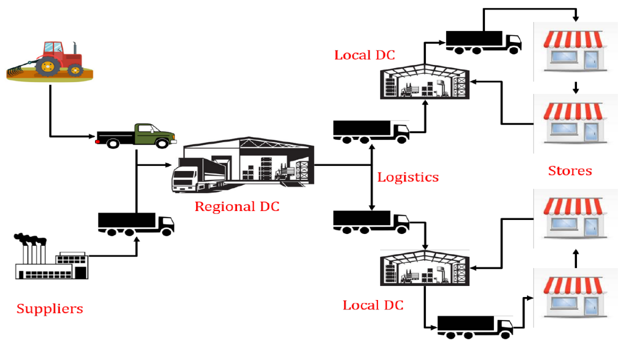

Fig. 21. Schematic of supply chain

ESL programs translate to Java source code which is the compiled and run using a standard Java compiler. The MIT Beer Game and a real-world supply chain were implemented using reinforcement learning and ESL agents. The real world case study is a large grocery retailer is a network of network of stores (S), local distribution centers (LDC), regional distribution centers (RDC), and retailers (R) as shown in Figure 21. The end points of this network sell thousands of products to its customers, local distribution centers replenish the products to the stores, and regional distribution centers procure products from retailers and supply to local distribution centers. All elements in this network are connected through a fleet of trucks. The goal of the authors is to regulate the availability of the entire product range in each store.

The initial tests of the implemented case study show that when using ESL the agents responses greatly improves. The authors conclude that when conducting a simulation using ESL it can improve the system performance, since it's possible to know the implications that each agent will have when using a proposed solution (evaluate the profit of a specific propose action or solution).

The article has great importance because the authors knowing the complexity, uncertainty, and lack of knowledge when systems are executed into a dynamic software ecosystem, proposes a feasible solution using MARL that allows systems to use artificial intelligence to react to changes giving a rapid and reliable response to a certain action, the authors also demonstrate that MARL can be used to address intra and inter system-complexity by allowing general systems to achieve an optimal behavior through dynamic learning.

4.4 Others communication Trends

In the analyzed literature the following communication trends are used but they're less common:

- **Web services:** Schroeder focuses his work [81] on creating an architecture using the CPS and the DT technology, to access system data-information through different local data repositories through web services (RESTful). Finally, the authors use the Augmented Reality (AR) technology to show the results.

- **Decision tree:** Talkhestani bases his research [86] on verifying the consistency (fidelity) of a digital twin based on a car manufacturing system using anchor points. The authors realized a cross-domain simulation to integrate the different models that form the system, the relationships between the different DTs of the system are controlled by a decision tree where changes like adding or removing a system component affect the performance of it.

5 Usage Trends

In this section, the result of the systematic research on Google Scholar with the topic of digital twins will be presented, with the aim of knowing the state-of-the-art of this new concept and being able to glimpse possible areas of study.

5.1 Focused areas

Through the analysis of the 32 case studies, it was possible to observe that it's feasible to apply the technology of digital twins in the following fields:

- Manufacturing (M), according to Kritzinger [50], a DT represents the life cycle of a system, which allows it to be able to digitally present all the elements of the system, from the most basic to the most advanced characteristics. This field is the most used in the analyze literature, that's why there are several uses of the DTs in it, the most relevant articles in the area will be briefly described below. Bottani in his work [17] creates a DT prototype that represents an automatic guided vehicle (AGV), supported by the CPS technology. In particular, the authors focus on the development of a DT of a cyber-physical automated guided vehicle (CGV) to solve the typical resources handling problem that occurs in a Job-Shop small fabrication system, to perform the system simulation, the authors use the DES simulation

Vachálek [93] creates a DT of a pneumatic cylinder production line, with the aim of optimizing the production process and performing proactive maintenance on it, the creates DT works through an interface, through which it obtains the data from physical twin. To create the simulated production part didactic stations were created using the software FESTO, the DM was created with the Plant Simulation (PS) tool. A genetic algorithm was used to optimize the production process, specifically, the DT realized simulations to optimize the created model.

Zhang [100] creates a DT of a hollow glass manufacturing system. The DT includes: the physical model of the system and obtains the data-information in real-time from its simulated physical twin, with the objective of modeling the system performance for the pre-production process, and it has the ability to make smart decisions.

Raileanu [70] creates a DT of a Shop Floor manufacturing system, the system work through a closed circuit. Assets are transported on pallets through the conveyor belt (controlled by its corresponding PLC). The digital twin is in charge of monitoring the behavior of the conveyor belt, once it receives enough data (travel intervals, time responses, etc.) realizes forecasts to optimize the behavior of the system. Other examples of the manufacturing area can be seen in [45, 82, 79, 4, 40, 13, 59, 101, 7, 86, 62, 5, 71, 20, 15, 51, 35]

- Aeronautics and Space (AS), Tuegel [90] proposes the reengineering of one aircraft structural life prediction process, Rios [73] created a framework which facilitates the univocal connection between the flying twin (aircraft) identified by an ID, and its DT; and Majumdar [60] examines how the multi-physical environment (electric field) of the airplanes materials (fiber reinforced polymers) causes microstructural changes and can affect their structural performance.

- Informatics (I), in this field, the DTs are used to create control architectures based on this technology. Yun [99] proposes a large-scale architecture, containing a DT-based platform, the platform includes an interface in charge of collaborative interactions inside the system, a responsive communication middleware, and is used to develop a reliable system based on ADAS (Advanced Driver Assistance System). The authors use international standards for their platform, called uDiT (universal Digital Twin platform), and presents the next elements: a middleware supported by OMG (Object Management Group) and DDS (Data Distribution Service) tools; besides allows co-simulation through FMI, and presents gateway methods. Others examples that belong to the informatics area and that have been previously described can be seen in [81, 3, 66, 44, 52, 75, 23].

5.2 Key enabling technologies

In agreement with Kritzinger [50], the DTs presents different levels of integration, and each one presents particular characteristics. Due to this, there isn't a generic tool to develop it, instead is possible to use a wide range of technologies The used technologies to execute a DT can be classified into the following classes:

- Simulation methods: in the analyzed literature it was possible to observe diverse simulation methods, the most widely used are: DES (discrete-event simulation), 3D simulation, and FEM (finite-element method) simulation.

- Simulation software: as in the simulation methods, various simulation software could be observed in the analyzed literature, the most used are: Arena, Festo, and FlexSim.

5.3 Categorization methodology

Nowadays, already exists a few literature reviews of the Digital Twin technology [64, 50], however, is still necessary to address some issues such as the synergy between DT and MAS. The main goal of this paper is to provide a bibliographic review of the existing literature, which allows the reader to understand the operation and main concepts of technology, as well as present the most important trends nowadays. And also present the functioning and main characteristics of the synergy between DT and MAS.

The authors employ the search engines Google Scholar and Scopus to collect papers published between 2010 and 2020, that include the term "Digital Twin" in the title, abstract, keywords, or in the content. Further, the authors selected the following keywords to perform the research: Digital Twin, Digital twin Manufacturing System, Digital Twin with Multi-Agent Systems, and Digital Twin Society.

No.	Ref	Year	Field	Usage	Level of Integration	Simulation	Simulation Software	Diagnostics Prognostics
1	[60]	2013	AS	Study the possible conduct of a structure during dynamic changes in its environment	DM	DT are simulations	N/A	Forecast of structural elements and their behavior in a dynamic environment
2	[74]	2016	AS	Aircraft design support	DM	iDMU	Dassault Systemes V6	N/A
3	[17]	2017	M	Automatic guidance vehicle simulation(AGV)	DM	DES	N/A	N/A
4	[79]	2017	M	Preserve the geometric features of a product	DM	N/A	N/A	N/A
5	[99]	2017	I	Platform to develop DT	DM	FMI	N/A	Model life prediction
6	[59]	2018	M	Digital model of a Micro Manufacturing Unit	DM	FlexSim	SolidWorks	N/A
7	[7]	2018	M	Reconditioning of a production system with a DT	DM	3D	Simumatik 3D	Malfunctions
8	[90]	2011	AS	Predict the structural life of an aircraft	DS	FEM and CFD	N/A	Structural deflections, temperatures, local damage and evolution of the material state
9	[82]	2016	M	Monitor the physical twin	DS	SDX	N/A	N/A
10	[81]	2016	I	Access DT data	DS	Augmented reality	Vuforia	DT values
11	[93]	2017	M	Model the behavior of a pneumatic cylinder production line	DS	Plant Simulation, OPC-UA	Festo	Proactive Maintenance
12	[40]	2018	M	Create a digital model of a bending beam machine	DS	FEM	N/A	N/A
13	[86]	2018	M	Consistency check on a DT through anchor points	DS	Cross-domain simulation	Festo	System behavior when a new component is added
14	[45]	2010	M	Decision-support system	DT	DES	N/A	Preventive Maintenance
15	[100]	2017	M	Design and optimization of a system through DT	DT	3D	N/A	System operation
16	[3]	2017	I	Integrate cloud support to a CPS	DT	N/A	N/A	N/A
17	[4]	2017	M	CP MAS for autonomous quality control	DT	N/A	N/A	System behaviour, failures
18	[44]	2018	I	Dynamic Co-Simulation based on DT	DT	N/A	Matlab, Open Modelica	System behavior when a new component is added
19	[101]	2018	M	DT of a production line	DT	3D	N/A	System behavior
20	[13]	2019	M	DT of a Body-in-white production system	DT	3D	DelmiaV5	System behavior when a new component is added

21	[62]	2019	M	FMU Simulation with DT	DT	DES	Matlab	Energy consumption
22	[70]	2019	M	DT of a Shop Floor manufacturing system	DT	N/A	N/A	System behavior, failures
23	[5]	2019	M	Creation of a communication middleware	DT	3D	FlexSim	N/A
24	[71]	2019	M	SLADTA architecture extension	DT	N/A	Tecnomatix Plant Simulation	N/A
25	[20]	2019	M	Generic framework for model integration	DT	DES	Rockwell-Arena	System operation, failures
26	[15]	2019	M	DT to manage a semi-continuos smart manufacturing system	DT	3D	N/A	System behavior
27	[52]	2019	I	CP MAS for farm management	DT	N/A	N/A	System behavior, forecasting
28	[66]	2019	I	MAS for collaborative learning and prognostics	DT	N/A	N/A	Collaborative prognostics
29	[51]	2019	M	MAS for distributed Job-Scheduling	DT	N/A	Witness 14	System behavior, scheduling
30	[35]	2019	M	CP MAS for ARM	DT	N/A	N/A	System behavior when a new element is added, failures
31	[75]	2019	I	MAS for evaluation of EV Charging Systems	DT	FMI	Simulation X, NetLogo	System behavior
32	[23]	2020	I	Language support for MARL	DT	N/A	ESL	System behavior, failures

Field: M=Manufacturing, AS= Aeronautics and Space, I=Informatics, R=Robotics

Table 2. Results of the classification method (method improved and adapted from Negri [64] and Kritzinger [50])

The works of Negri, and Kritzinger were used as foundation to create this paper, the categorization methodologies of these authors were analyzed for later be adapted and improved. Negri in his work [64] at 2017, analyzes the existing literature related to DT technology, classifies the analyzed literature according to: its field (Manufacturing, Aeronautics and Space, Robotics and Informatics), diagnosis and prognostics, type of simulation and simulation software. Kritzinger [50] in 2018, performs an unequivocal literature review of the DT to classify the analyzed publications in accordance with their level of integration of the DT (Digital Twin, Digital Shadow and Digital Model). By analyzing the literature reviews realized out by these authors, it was possible to determine that performing a synergy of both classification methods in the analyzed literature, allows knowing the most important elements in each one of the papers and perform a deeper analysis of it, at the same time that it improves the reader's understanding of each article purpose. It's important to highlight that since the classification method is based on the work of these authors, the results obtained from it will contrasted with the results achieved by the authors in their works.

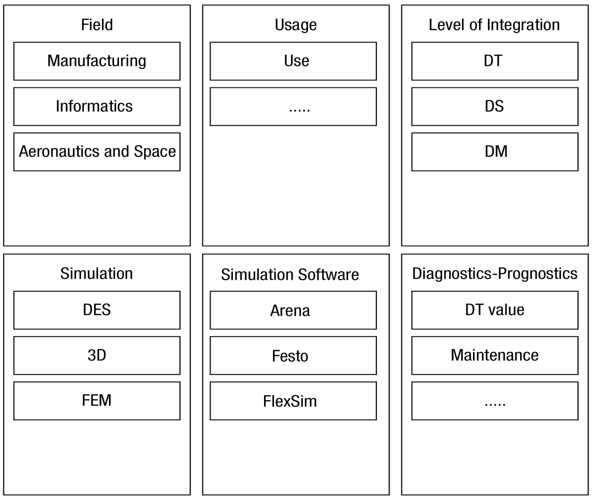

Fig. 22. Categorization Method

The publications found were analyzed according to their content and categorized through its main features, in Fig. 22. readers can visually perceive the elements that form the categorization method, for better compression. First, the field of application of the publications: M (Manufacturing), AS (Aeronautics and Space, I (Informatics) was determined. Next, the main use or purpose of each article is detailed; then, the level of integration presented by the digital twin that is detailed in each of the articles was established; after that, the simulation technology and the simulation software used were presented; finally, it was exposed if the articles made any diagnosis or prognosis to know if the created DT was used to develop future forecasts and to know the contribution of these forecasts to the operation of the product or system. Therefore, the categorization method was used, the results are shown in Table 3.

5.4 Discussion

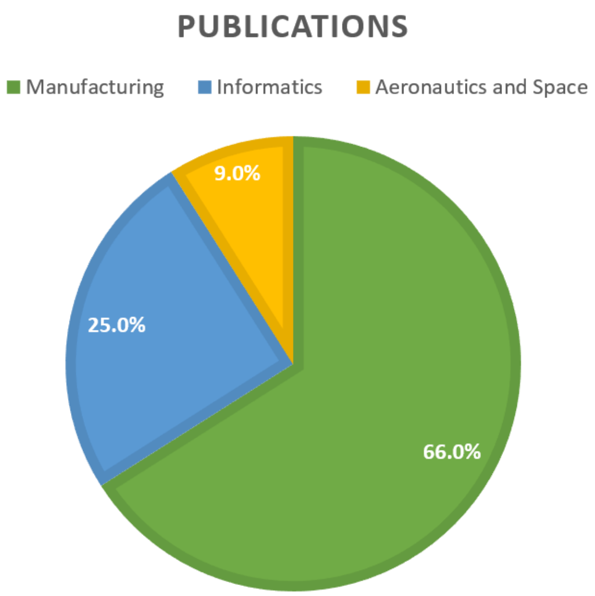

Fig. 23. Publication fields

Negri in his research [64], get as results that the majority of most of the analyzed literature belong to the area of aeronautics and space (69 percent), in which the articles are based on monitoring the operation of a system and the reliability of the created model. In the manufacturing area (19 percent) the DT technology is used to monitor the lifecycle of a system and optimize its design. In the robotics field (8 percent) the DT is used for virtual commissioning. Finally, in the informatics area (4 percent), the DT is used to IoT lifecycle management.

In the realized research the majority (66 percent) of the analyzed publications are categorized in the manufacturing field because this is the area where the DT technology presents more uses and advantages (e.g test, optimize, and update the production system). The informatics area has 25 percent, the use of DT in this field is based on the creation of control architectures; in the field of aeronautics and space (9 percent), the authors use DT to control processes concerning the life cycle of an aircraft. The results can be viewed in the Fig. 23.

It's possible to observe that a drastic change has occurred in the main area of use of DT technology since Negri's work in 2017 until the current research in 2020, with the main area currently being the manufacturing one, this change occurs because over the years numerous uses of the DT technology have been developed in this area, these uses allow owners of manufacturing companies to have numerous advantages in using them to make improvements and forecast the system behavior throughout the life cycle of it.

It's also possible to observe that the use of DT in the AS area has progressed to support the control and monitoring of aircraft. Regarding the informatics field, the use of DT technology has been largely developed through the creation of control architectures. Is also important to highlight that the robotics field was not included in the research since the articles developed in this field present a very low number of citations and minor importance for the proposed goal.

Regarding the level of integration, Kritzinger in his work [50], get as most used level the Digital Shadow (35 percent), followed by Digital Model (28 percent) and with a lower percentage the Digital Twin (18 percent), having a 19 percent of publications of indeterminate level of integration.

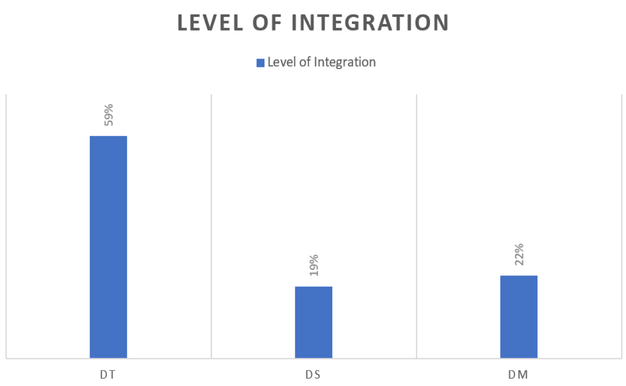

Fig. 24. Level of Integration

In the research, the major part of the papers are categorized as Digital Twin (59 percent) and Digital Model (22 percent), with a lower percentage we have the classification Digital Shadow (19 percent). Although all articles use the term Digital Twin within their content, not all of them present the bi-directional exchange of data among the DT and the physical twin. The publications that present data exchange do so through the main communication trends, addressed in a previous section, however, only the publications [71, 15, 86, 92, 20, 5, 66, 52, 44, 4, 51, 35, 75, 23] present communication between DTs of the same system. The results of the categorization can be seen in Fig. 24.

It can be noted that the most popular level of integration since Kritzinger's work has changed, currently being Digital Twin, because the majority of publications since 2017 allow a bi-directional data exchange among physical and digital twins, and also between digital twins that belong to the same system. It can be also noticed that all the analyzed publications present a concrete level of integration.

In the matter of the usages or purposes of the publications according to the research of Negri [64], the most popular use of DT technology in the analyzed publications by the authors, is to monitor the life cycle of a system to predict behavior and detect failures. In the reviewed publications the purposes are very diverse, however, many of the authors use DT technology to test the operation of a manufacturing system, to realize an improvement or update it, to collaborate in decision making, and also to develop control architectures, these uses are the most popular today because they are the most beneficial ones for saving resources, money and processing time.

The simulation technology and simulation software are also very variable, in Negri's work the most common ones are FEM and Matlab respectively, in Kritzinger's work it's only possible to identify the most widely used simulation technology: DES. Among the more frequently used today are the following: for simulation technology (DES, 3D, and FEM), and for simulation software (Arena, Festo, and FlexSim). Regarding simulation technology, the mentioned technologies are the most used for the multiple benefits they present; DES highly accuracy, greater flexibility, and model validation [47]; 3D allows the visuospatial appreciation and understanding of the complex architecture of the system; FEM increased accuracy, easy adaptability, and is faster and less expensive.

Concerning simulation software, simulation software has been diversifying over time, the authors of the analyzed publications opted for the most popular software due to their reliability, low use of resources, speed and cost, ease of use, and because most of them allow creating virtual prototypes.

With reference to diagnostics and prognostics in Negri's work, the most commonly used is predictive maintenance for fault detection. In the analyzed papers the increasingly used are: predictive or proactive maintenance to forecast system operation and detect failures. It can be noted that in the works analyzed by Negri it's not possible to forecast the operation of the system when a new element is included in it, because this function has been developed in recent years.

6 Conclusions, contributions, challenges, and further work

6.1 Conclusions

At the end of the analysis, it's possible to conclude that, the majority of existing case studies focus on the manufacturing field, because it's in which the greatest number of advantages presented by the DTs are currently known. The publications made in recent years show progress in terms of communication, because they allow communication between DTs of the same system, which wasn't allowed it before 2017. It's also important to highlight that in the last years the case-studies present the highest level of integration (DT), which allows bi-directional exchange of data among the physical twin and digital twin, in the previous years (before 2017) the common integration levels were the lowest (DM and DS). Also, in some case studies especially in those published in recent years (since 2017), multi-agent systems (MAS) technology is used to control the decision making of digital twins and provide them with intelligence through artificial intelligence.

Finally, it's also possible to conclude that nowadays there isn't a unified or generic DT modeling method, neither exists generics benchmarks that can be used in all development areas of the DT technology, however, there are some recent works based on a chemical process where the notorious Tennessee Eastman (TE) benchmark process [27] is utilized to evaluate the created DT models [57, 41].

6.2 Contributions

The principal contributions of this paper are summarized below:

- Through an analysis of the existent literature using a computer science approach, the papers clarifies the DTs concept, mains aspects and characteristics.
- It outlines the main communication trends, between physical twin and digital twin, also between digital twins of a same system and digital twins with an external system.
- It highlights the synergy between digital twin and MAS system technologies, introducing the most meaningful case studies regarding this issue.

6.3 Challenges

Although DT technology has achieved substantial advance, especially in the publications made in last years, it's necessary to realize future research, especially in the following topics:

1. Integrate the DT during the entire life cycle of an object or manufacturing system.
2. Use digital twins to test the operation of new production control methods (e.g. Synchro-push production), especially in industrial environments.
3. Employ the DTs to collaborate in the resolution of the main problems in the field of manufacturing (scheduling, material handling).

4. Utilize Artificial intelligence (AI) and Multi-Agent Systems to optimize decision-making in the DTs, and to realize more accurate predictions.
5. Use industry 4.0 standards to optimize communication between digital twins of the same system.
6. Achieve an effective cyber-physical fusion, including the several technologies that a CPS needs to properly work.
7. Create a unified method for creating digital twin models.

6.4 Further Work

As future work, the authors aim to contribute to the development of challenges 4 and 5, by developing a MAS support framework, with the main goal of optimizing decision-making and allowing communication between intelligent digital twins that belong to the same or different ecosystems.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] David Abeijón, Francesco Soriguera, and Leif Thorson. Fusión de datos para obtención de tiempos de viaje en carretera, 2007.

[3] Kazi Masudul Alam and Abdulmotaleb El Saddik. C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE Access*, 5:2050–2062, 2017.

[4] Narjes Alaya, Baudouin Dafflon, Nejib Moalla, and Yacine Ouzrout. A self-adaptative cps-agent based quality control platform for industry 4.0. *University Claude Bernard Lyon, University Lumire Lyon 2*, 2017.

[5] Pascal André, Fawzi Azzi, and Olivier Cardin. Heterogeneous communication middleware for digital twin based cyber manufacturing systems. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 146–157. Springer, 2019.

[6] Erhan Batuhan Arisoy, Guannan Ren, Erva Ulu, Nurcan Gecer Ulu, and Suraj Musuvathy. A data-driven approach to predict hand positions for two-hand grasps of industrial objects. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V01AT02A067–V01AT02A067. American Society of Mechanical Engineers, 2016.

[7] Mikel Ayani, Maria Ganebäck, and Amos HC Ng. Digital twin: Applying emulation for machine reconditioning. *Procedia CIRP*, 72:243–248, 2018.

[8] Manas Bajaj, Bjorn Cole, and Dirk Zwemer. Architecture to geometry-integrating system models with mechanical design. In *AIAA SPACE 2016*, page 5470. 2016.

[9] D. Bakken. Middleware,https://www.eecs.wsu.edu/ bakken/middleware.pdf, 2003.

[10] Kshitij Bakliwal, Maharshi Harshadbhai Dhada, Adria Salvador Palau, Ajith Kumar Parlikad, and Bhupesh Kumar Lad. A multi agent system architecture to implement collaborative learning for social industrial assets. *IFAC-PapersOnLine*, 51(11):1237–1242, 2018.

[11] Yuri Bazilevs, X Deng, A Korobenko, F Lanza di Scalea, MD Todd, and SG Taylor. Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data. *Journal of Applied Mechanics*, 82(9), 2015.

[12] Brent Bielefeldt, Jacob Hochhalter, and Darren Hartl. Computationally efficient analysis of sma sensory particles embedded in complex aerostructures using a substructure approach. In *ASME 2015 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*. American Society of Mechanical Engineers Digital Collection, 2015.

[13] Florian Biesinger, Davis Meike, Benedikt Kraß, and Michael Weyrich. A digital twin for production planning based on cyber-physical systems: A case study for a cyber-physical system-based creation of a digital twin. *Procedia CIRP*, 79:355–360, 2019.

[14] Torsten Blochwitz, Martin Otter, Martin Arnold, Constanze Bausch, Christoph Clauss, Hilding Elmqvist, Andreas Junghanns, Jakob Mauss, Manuel Monteiro, Thomas Neidhold, et al. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th International Modelica Conference*, pages 105–114. Linköping University Press, 2011.

[15] Theodor Borangiu, Ecaterina Oltean, Silviu Răileanu, Florin Anton, Silvia Anton, and Iulia Iacob. Embedded digital twin for arti-type control of semi-continuous production processes. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 113–133. Springer, 2019.

[16] Stefano Borgo. An ontological approach for reliable data integration in the industrial domain. *Computers in Industry*, 65(9):1242–1252, 2014.

[17] E Bottani, A Cammardella, T Murino, and S Vespoli. From the cyber-physical system to the digital twin: the process development for behaviour modelling of a cyber guided vehicle in m2m logic. *XXII Summer School Francesco TurcoIndustrial Systems Engineering*, pages 1–7, 2017.

[18] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*, pages 183–221. Springer, 2010.

[19] Arquimedes Canedo. Industrial iot lifecycle via digital twins. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, page 29. ACM, 2016.

[20] Olivier Cardin, Pierre Castagna, Daniel Couedel, Christophe Plot, Julien Launay, Nadine Allanic, Yannick Madec, and Stéphanie Jegouzo. Energy-aware resources in digital twin: The case of injection moulding machines. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 183–194. Springer, 2019.

[21] Sang Su Choi, Tae Hyuck Yoon, and Sang Do Noh. Xml-based neutral file and plm integrator for ppr information exchange between heterogeneous plm systems. *International Journal of Computer Integrated Manufacturing*, 23(3):216–228, 2010.

[22] JCH Christopher. Watkins and peter dayan. *Q-Learning. Machine Learning*, 8(3):279–292, 1992.

[23] Tony Clark, Balbir Barn, Vinay Kulkarni, and Souvik Barat. Language support for multi agent reinforcement learning. In *Proceedings of the 13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference*, pages 1–12, 2020.

[24] Jim Davis, Thomas Edgar, James Porter, John Bernaden, and Michael Sarli. Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Computers & Chemical Engineering*, 47:145–156, 2012.

[25] W Edwards Deming. *Out of the Crisis*. MIT press, 2018.

[26] GE DIGITAL. The digital twin: Compressing time to value for digital industrial companies. *Technical Report*, 2017.

[27] James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

[28] ESI:2019. Esi-iti. simulationx 4.0.s, 2019(accessed on 22 July 2019).

[29] Peter Fritzson and Peter Bunus. Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation. In *Proceedings 35th Annual Simulation Symposium. SS 2002*, pages 365–380. IEEE, 2002.

[30] Thomas Gabor, Lenz Belzner, Marie Kiermeier, Michael Till Beck, and Alexander Neitz. A simulation-based architecture for smart cyber-physical systems. In *2016 IEEE International Conference on Autonomic Computing (ICAC)*, pages 374–379. IEEE, 2016.

[31] Marco Garetti, Luca Fumagalli, and Elisa Negri. Role of ontologies for cps implementation in manufacturing. *Management and Production Engineering Review*, 6(4):26–32, 2015.

[32] Edward Glaessgen and David Stargel. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, page 1818, 2012.

[33] André Glória, Francisco Cercas, and Nuno Souto. Design and implementation of an iot gateway to create smart environments. *Design and implementation of an IoT gateway to create smart environments*, pages 568–575, 2017.

[34] Brian Gockel, Andrew Tudor, Mark Brandyberry, Ravi Penmetsa, and Eric Tuegel. Challenges with structural life forecasting using realistic mission profiles. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, page 1813, 2012.

[35] VI Gorodetsky, SS Kozhevnikov, D Novichkov, and Peter O Skobelev. The framework for designing autonomous cyber-physical multi-agent systems for adaptive resource management. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, pages 52–64. Springer, 2019.

[36] Vladimir Gorodetsky, Petr Skobelev, and Vladimır Marik. System engineering view on multi-agent technology for industrial applications: Barriers and prospects. *Cybernetics and Physics*, 9(1):13–30, 2020.

[37] Michael Grieves. Digital twin: Manufacturing excellence through virtual factory replication. *White paper*, pages 1–7, 2014.

[38] Michael Grieves and John Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary perspectives on complex systems*, pages 85–113. Springer, 2017.

[39] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[40] Sebastian Haag and Reiner Anderl. Digital twin–proof of concept. *Manufacturing Letters*, 15:64–66, 2018.

[41] Rui He, Guoming Chen, Che Dong, Shufeng Sun, and Xiaoyu Shen. Data-driven digital twin technology for optimized control in process systems. *ISA transactions*, 95:221–234, 2019.

[42] Stijn Heymans, Li Ma, Darko Anicic, Zhilei Ma, Nathalie Steinmetz, Yue Pan, Jing Mei, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, et al. Ontology reasoning with large data repositories. In *Ontology Management*, pages 89–128. Springer, 2008.

[43] ECMA International. The json data interchange syntax. *Standard ECMA-404,2nd edn*, 2017.

[44] Tobias Jung, Payal Shah, and Michael Weyrich. Dynamic co-simulation of internet-of-things-components using a multi-agent-system. *Procedia CIRP*, 72:874–879, 2018.

[45] Botond Kádár, A Lengyel, László Monostori, Y Suginishi, András Pfeiffer, and Y Nonaka. Enhanced control of complex production structures by tight coupling of the digital and the physical worlds. *CIRP annals*, 59(1):437–440, 2010.

[46] Henning Kagermann, Johannes Helbig, Ariane Hellinger, and Wolfgang Wahlster. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.

[47] Jonathan Karnon and Hossein Haji Ali Afzali. When to use discrete event simulation (des) for the economic evaluation of health technologies? a review and critique of the costs and benefits of des. *Pharmacoeconomics*, 32(6):547–558, 2014.

[48] Hazem Kaylani and Anas M Atieh. Simulation approach to enhance production scheduling procedures at a pharmaceutical company with large product mix. *Procedia CIRP*, 41:411–416, 2016.

[49] Edward M Kraft. The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge. In *54th AIAA Aerospace Sciences Meeting*, page 0897, 2016.

[50] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018.

[51] Sandeep Kumar, Bhupesh Kumar Lad, Maharshi Harshadbhai Dhada, and Kshitij Bakliwal. Distributed job scheduling using multi-agent system. *Proceedings of the International Conference on Industrial Engineering and Operations Management Bangkok, Thailand*, 2019.

[52] Vladimir Laryukhin, Petr Skobelev, Oleg Lakhin, Sergey Grachev, Vladimir Yalovenko, and Olga Yalovenko. The multi-agent approach for developing a cyber-physical system for managing precise farms with digital twins of plants. *CYBERNETICS AND PHYSICS*, 8(4):257–261, 2019.

[53] J Lee, HA Kao, and S Yang. Service innovation and smart analytics for industry 4.0 and big data environment. procedia cirp, 16, 3–8, 2014.

[54] Jay Lee, Behrad Bagheri, and Hung-An Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, 3:18–23, 2015.

[55] Jay Lee, Edzel Lapira, Behrad Bagheri, and Hung-an Kao. Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing letters*, 1(1):38–41, 2013.

[56] Christoph Legat, Christian Seitz, Steffen Lamparter, and Stefan Feldmann. Semantics to the shop floor: towards ontology modularization and reuse in the automation domain. *IFAC Proceedings Volumes*, 47(3):3444–3449, 2014.

[57] Weijun Li, Sai Gu, Xiangping Zhang, and Tao Chen. A pattern matching and active simulation method for process fault diagnosis. *Industrial & Engineering Chemistry Research*, 2020.

[58] Zheng Liu, Norbert Meyendorf, and Nezih Mrad. The role of data fusion in predictive maintenance using digital twin. In *AIP Conference Proceedings*, volume 1949, page 020023. AIP Publishing, 2018.

[59] Mika Lohtander, Niko Ahonen, Minna Lanz, Juho Ratava, and Jarno Kaakkunen. Micro manufacturing unit and the corresponding 3d-model for the digital twin. *Procedia Manufacturing*, 25:55–61, 2018.

[60] Prasun K Majumdar, Mohammad FaisalHaider, and Kenneth Reifsnider. Multi-physics response of structural composites and framework for modeling using material geometry. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1577, 2013.

[61] Murali Mani, Dongwon Lee, and Richard R Muntz. Semantic data modeling using xml schemas. In *International Conference on Conceptual Modeling*, pages 149–163. Springer, 2001.

[62] Elisa Negri, Luca Fumagalli, Chiara Cimino, and Marco Macchi. Fmu-supported simulation for cps digital twin. *Procedia Manufacturing*, 28:201–206, 2019.

[63] Elisa Negri, Luca Fumagalli, Marco Garetti, and Letizia Tanca. Requirements and languages for the semantic representation of manufacturing systems. *Computers in Industry*, 81:55–66, 2016.

[64] Elisa Negri, Luca Fumagalli, and Marco Macchi. A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

[65] Oracle. Digital twins for iot applications: a comprehensive approach to implementing iot digital twins http://www.oracle.com/us/solutions/internetofthings/digital-twins-for-iot-appswp-3491953.pdf. *Oracle White Paper,January 2017*, 2017.

[66] Adrià Salvador Palau, Maharshi Harshadbhai Dhada, Kshitij Bakliwal, and Ajith Kumar Parlikad. An industrial multi agent system for real-time distributed collaborative prognostics. *Engineering Applications of Artificial Intelligence*, 85:590–606, 2019.

[67] IEC PAS. 62424 ?specification for representation of process control engineering requests in p&i diagrams and for data exchange between p&id tools and pce-cae tools ? *VDE-Verlag GmbH, Berlin*, 2006.

[68] Michael J Pratt. Introduction to iso 10303?the step standard for product data exchange. *Journal of Computing and Information Science in Engineering*, 1(1):102–103, 2001.

[69] Qinglin Qi and Fei Tao. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *Ieee Access*, 6:3585–3593, 2018.

[70] Silviu Răileanu, Theodor Borangiu, Nick Ivănescu, Octavian Morariu, and Florin Anton. Integrating the digital twin of a shop floor conveyor in the manufacturing control system. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 134–145. Springer, 2019.

[71] AJH Redelinghuys, K Kruger, and Anton Basson. A six-layer architecture for digital twins with aggregation. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 171–182. Springer, 2019.

[72] Kenneth Reifsnider and Prasun Majumdar. Multiphysics stimulated simulation digital twin methods for fleet management. In *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1578, 2013.

[73] José Ríos, Juan Carlos Hernández, Manuel Oliva, and Fernando Mas. Product avatar as digital counterpart of a physical individual product: Literature review and implications in an aircraft. In *ISPE CE*, pages 657–666, 2015.

[74] José Ríos, Fernando Mas Morate, Manuel Oliva, and Juan Carlos Hernández. Framework to support the aircraft digital counterpart concept with an industrial design view. *International Journal of Agile Systems and Management*, 9(3):212–231, 2016.

[75] Tobias Rodemann, Tom Eckhardt, René Unger, and Torsten Schwan. Using agent-based customer modeling for the evaluation of ev charging systems. *Energies*, 12(15):2858, 2019.

[76] Roland Rosen, Georg Von Wichert, George Lo, and Kurt D Bettenhausen. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3):567–572, 2015.

[77] G. Rzevski and P. Skobelev. Managing complexity. *WIT Press*, 2014.

[78] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management*, pages 1–9. IEEE, 2008.

[79] Benjamin Schleich, Nabil Anwer, Luc Mathieu, and Sandro Wartzack. Shaping the digital twin for design and production engineering. *CIRP Annals*, 66(1):141–144, 2017.

[80] Michael Schluse and Juergen Rossmann. From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems. In *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–6. IEEE, 2016.

[81] Greyce Schroeder, Charles Steinmetz, Carlos Eduardo Pereira, Ivan Muller, Natanael Garcia, Danubia Espindola, and Ricardo Rodrigues. Visualising the digital twin using web services and augmented reality. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pages 522–527. IEEE, 2016.

[82] Greyce N Schroeder, Charles Steinmetz, Carlos E Pereira, and Danubia B Espindola. Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine*, 49(30):12–17, 2016.

[83] Mike Shafto, Mike Conroy, Rich Doyle, Ed Glaessgen, Chris Kemp, Jacqueline LeMoigne, and Lui Wang. Modeling, simulation, information technology & processing roadmap. *National Aeronautics and Space Administration*, 2012.

[84] Christopher Simpkins and Charles Isbell. Composable modular reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4975–4982, 2019.

[85] R. F. Sudarsan. A product information modeling framework for product lifecycle management. *Computer-aided design*, 37(13):1399–1411, 2001.

[86] Behrang Ashtari Talkhestani, Nasser Jazdi, Wolfgang Schloegl, and Michael Weyrich. Consistency check to synchronize the digital twin of manufacturing automation based on anchor points. *Proc. CIRP*, 72:159–164, 2018.

[87] Fei Tao, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9-12):3563–3576, 2018.

[88] Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415, 2018.

[89] Eric Tuegel. The airframe digital twin: some challenges to realization. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, page 1812, 2012.

[90] Eric J Tuegel, Anthony R Ingraffea, Thomas G Eason, and S Michael Spottswood. Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, 2011, 2011.

[91] INTELLIGENCE BY AM TURING. Computing machinery and intelligence-am turing. *Mind*, 59(236):433, 1950.

[92] Jumyung Um, Stephan Weyer, and Fabian Quint. Plug-and-simulate within modular assembly line enabled by digital twins and the use of automationml. *IFAC-PapersOnLine*, 50(1):15904–15909, 2017.

[93] Ján Vachálek, Lukás Bartalskỳ, Oliver Rovnỳ, Dana Šišmišová, Martin Morháč, and Milan Lokšík. The digital twin of an industrial production line within the industry 4.0 concept. In *2017 21st International Conference on Process Control (PC)*, pages 258–262. IEEE, 2017.

[94] Paul Valckenaers. Arti reference architecture–prosa revisited. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 1–19. Springer, 2018.

[95] Paul Valckenaers and Hendrik Van Brussel. Intelligent products: Intelligent beings or agents? In *International Conference on Information Technology for Balanced Automation Systems*, pages 295–302. Springer, 2008.

[96] Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, and Patrick Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in industry*, 37(3):255–274, 1998.

[97] Hai-Kun Wang, Robert Haynes, Hong-Zhong Huang, Leiting Dong, and Satya N Atluri. The use of high-performance fatigue mechanics and the extended kalman/particle filters, for diagnostics and prognostics of aircraft structures. *CMES: Computer Modeling in Engineering & Sciences*, 105(1):1–24, 2015.

[98] Jennifer Whyte, N Bouchlaghem, A Thorpe, and R McCaffer. From cad to virtual reality: modelling approaches, data exchange and interactive 3d building design tools. *Automation in construction*, 10(1):43–55, 2000.

[99] Seongjin Yun, Jun-Hong Park, and Won-Tae Kim. Data-centric middleware based digital twin platform for dependable cyber-physical systems. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 922–926. IEEE, 2017.

[100] Hao Zhang, Qiang Liu, Xin Chen, Ding Zhang, and Jiewu Leng. A digital twin-based approach for designing and multi-objective optimization of hollow glass production line. *IEEE Access*, 5:26901–26911, 2017.

[101] Yu Zheng, Sen Yang, and Huanchong Cheng. An application framework of digital twin and its case study. *Journal of Ambient Intelligence and Humanized Computing*, 10(3):1141–1153, 2018.

Table Index

Figure Index