MADTwin : a framework for multi-agent digital twin development: smart warehouse case study

Reference:

Marah Hussein, Challenger Moharram.- MADTwin : a framework for multi-agent digital twin development: smart warehouse case study
Annals of mathematics and artificial intelligence - ISSN 1573-7470 - (2023), p. 1-31
Full text (Publisher's DOI): https://doi.org/10.1007/S10472-023-09872-Z
To cite this reference: https://hdl.handle.net/10067/1977250151162165141

MADTwin: A Framework for Multi-agent Digital Twin Development: Smart Warehouse Case Study

Hussein Marah[1*] and Moharram Challenger[1]

[1*]Department of Computer Science, University of Antwerp & Flanders Make Strategic Research Center, Flanders, Belgium.

*Corresponding author(s). E-mail(s): hussein.marah@uantwerpen.be;
Contributing authors: moharram.challenger@uantwerpen.be;

Abstract

A Digital Twin (DT) is a frequently updated virtual representation of a physical or a digital instance that captures its properties of interest. Incorporating both cyber and physical parts to build a digital twin is challenging due to the high complexity of the requirements that should be addressed and satisfied during the design, implementation and operation. In this context, we introduce the **MADTwin** (Multi-Agent Digital Twin) framework driven by a Multi-agent Systems (MAS) paradigm and supported by flexible architecture and extendible upper ontology for modelling agent-based digital twins. A comprehensive case study of a smart warehouse supported by multi-robots has been presented to show the feasibility and applicability of this framework. The introduced framework powered by intelligent agents integrated with enabler technologies enabled us to cope with parts of the challenges imposed by modelling and integrating Cyber-Physical Systems (CPS) with digital twins for multi-robots of the smart warehouse. In this framework, different components of CPS (robots) are represented as autonomous physical agents with their digital twin agents in the digital twin environment. Agents act autonomously and cooperatively to achieve their local goals and the objectives of the whole system. Eventually, we discuss the framework's strengths and identify areas of improvement and plans for future work.

Keywords: Digital Twin, Intelligent Agent, Multi-agent System, Cyber-Physical System

1 Introduction

The past few years have witnessed significant technological advancement as several technologies have emerged and improved remarkably. This progress has led to the era of digital transformation and smart manufacturing [54] propelled by the fourth industrial revolution (Industry 4.0) [38, 23]. Systems' requirements have been becoming higher and more complex than before due to the nature of large-scale and distributed systems such as Cyber-Physical Systems (CPS) and the Internet of Things (IoT) utilized to achieve this digitization. Heterogeneous components and sophisticated dynamic behaviour characterize these systems [47]. For this reason, old and conventional solutions and approaches are being replaced by intelligent methods to satisfy high criteria and attain desirable characteristics such as efficiency, accuracy, flexibility, reliability, etc.

CPS are distributed and heterogeneous systems comprising many interrelated parts and sub-components, making them inherit the system-of-systems (SoS) characteristics that increase their complexity. Complexity grows as more subsystems and sub-components are included. Implementing and designing CPS comes with a cost, such as dealing with a high level of complexity [45]. In addition, due to the lack of advanced technologies for integrating physical and digital parts [49], certification of the system under study (SUS) and its operations and performing testing, monitoring, validating, and analyzing were often conducted offline or in a real environment which is unsafe, and very expensive.

Despite all the existing technologies, achieving full integration [49], to map physical systems with their digital counterparts is still in progress. The primary motivation for achieving such integration is representing the physical world in a digital form as realistically as possible [18]. This digital-physical integration has the potential to revolutionize the way we design, monitor, maintain and improve the digital or the physical SUS.

A new generation of technology represented in digital twin [15], has emerged to cope with the challenges of integrating physical and digital parts. A digital twin is a virtual representation of a physical object or a system that establishes a real-time synchronization and a control loop between physical and digital instances. However, designing and deploying a digital twin for CPS is significantly challenging as several aspects and requirements should be considered. Fig. 1 highlights the key challenges and requirements.

However, current approaches lack modular, flexible, extendable capabilities that can provide digital twins with the desired level of intelligence and autonomy and integrate physical and digital parts conveniently. In this context, we use the agent paradigm to target part of these challenges and limitations, such as using a modular framework that can deploy both physical and digital parts effortlessly and flexibly [7], also modelling physical and digital parts to have autonomous capabilities to make them more intelligent to reach their goals; besides, representing heterogeneous components of physical and digital parts with a unified formalism; and finally having a reliable communication between physical and digital worlds by setting up fast, standard and reliable communication channels for real-time data and discrete events. Leveraging agent paradigm capabilities can help to design more flexible, robust, resilient, reliable and intelligent digital twins.

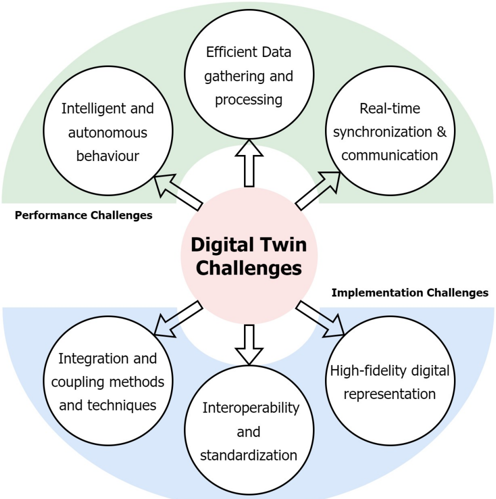

Fig. 1 Dimensions of Challenges in Digital Twin

Ultimately, the key aim and the core essence of this research is to explore the potential of agents by providing a framework based on an intelligent agent-based approach that combines agents' capabilities with other enabler technologies, such as IoT data-sensing for modeling digital twins.

From this perspective, we have some research questions that we will answer through this paper in particular:

1. What is the feasibility and applicability of the agent paradigm to tackle the issues of integrating cyber and physical parts of digital twins for CPS?;
2. To what extent the agent-based solutions can be utilized to provide autonomous, intelligent, pluggable, flexible, reliable, scalable, load-balanced and robust digital twin designs for CPS?
3. How could the enabler technologies, particularly IoT sensing and messaging, be integrated with agent technology to overcome agents' real-time limitations?

Originally, this work extends a previous work presented in [19]. In the previous work, we introduced our vision and the big picture of our conceptual idea. This paper builds on that work and provides a more elaborated framework supported by extended architecture and upper ontology; besides, we extended the case study to show the credibility of our approach. In a nutshell, this paper has four main contributions:

1. Providing a framework supported by general-purpose and extendable architecture for designing agent-based digital twins for CPS,
2. Defining modifiable and expandable upper ontology that guides practitioners to implement domain-specific intelligent agent-based digital twin solutions for a complex and distributed CPS; this ontology can serve as the cornerstone for achieving interoperability between different agent-based digital twins implementations, as it can be extended to domain-specific ontology.
3. Establishing reliable communication by utilizing two types of communication between physical and digital assets. The framework exploits the JADE MAS platform to design, build, and deploy physical and digital agents. Agents in both assets use standard agent communication. With this well-defined communication between physical and digital agents, we tackled the challenge of integrating and coupling physical and digital assets, as agent event-based communication is carried out efficiently with this integration. In addition, by utilizing IoT sensor technologies, we handled the challenge of having soft real-time synchronization between physical and digital assets and achieving appropriate data gathering from the physical asset.
4. Providing a development methodology that can be followed while designing an agent-based digital twin.

Finally, we showed a proof-of-concept of our framework with a case study by implementing mobile robots in a warehouse. Then, a comprehensive discussion of the pros and cons is highlighted to identify areas of research that will push forward the progress of our current research project.

2 Background

This section elaborates on the preliminary concepts used in this paper, including digital twins and intelligent agents.

2.1 Digital Twin

A digital twin is a new cutting-edge technology that has emerged in the new millennium [15]. Due to its potential capabilities, digital twin has become an interesting research topic in digital transformation, attracting many scientists and engineers from academia and industry as well [38, 14, 49]. Since the formulation of the technology and its initial concept, several researchers have tried to adopt and reform the origin concept according to their domain, from manufacturing to the aerospace industry and other disciplines [34]. Basically, a digital twin represents a physical instance or process that is intertwined with its conformable entity in the virtual world. An overview of the digital twin concept is depicted in Fig 2.

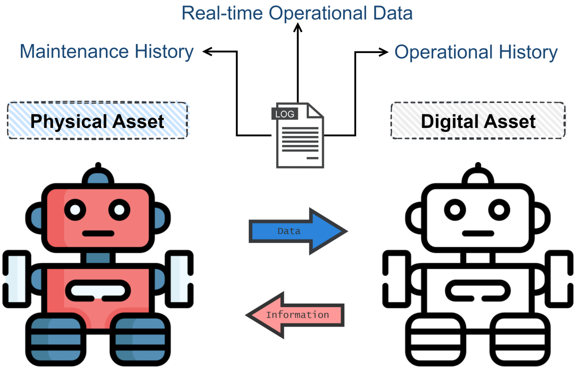

Fig. 2 Digital Twin Concept Overview Diagram

The enabling technologies, especially in the context of Industry 4.0 (e.g., IoT, cloud computing, big data, machine learning (ML), and sensors), facilitate the operation of integrating physical elements with their digital representatives and synchronizing their status inside a digital twin. Implementing a digital twin can offer many benefits, especially in manufacturing [40, 14]. As mentioned in the definition of the digital twin, a data connection between physical and digital parts is instantiated. This data stream can be quite valuable in the product life-cycle [41]. It can be leveraged for various purposes such as development, improvement, and management tasks like analyzing, inspecting, monitoring, predicting, and reasoning about the system's behaviour and making decisions accordingly to enhance the current system and avoid possible failures [49]. Above that, this data flow can be persisted and stored as historical data, which could be beneficial where applying learning mechanisms [40] on this data can improve the system's performance. In addition, integrating the physical and digital parts in the digital twin could boost the process of extending, modifying and customizing the physical design based on the insights gained from the collected data (e.g., operational real-time and historical data).

2.2 Intelligent Agents

The core element in Agent-based Modeling (ABM) approach and Multi-agent Systems (MAS) paradigm is the ***agent*** [51]. The agent approach provides powerful features to systems modelled with this technology. Agent's intelligence is formed and composed of the combination of several characteristics and properties that constitute agents' behaviours [51, 42]. For instance, agents are autonomous entities, which means they operate without external intervention to reach their internal goals. They can decide to collaborate, interact, negotiate and communicate with other agents directly or indirectly if a mutual benefit can be achieved at the level of the agent itself or the level of the whole system [26]. Also, agents are dynamic social entities because they are located in a specific environment, and they need, in some cases, to interact with external entities (e.g., humans or third party agents) to achieve the global goals of the system or their own goals that cannot be reached individually. In addition, agents are reactive units; they perceive the environment and surroundings and respond promptly to changes at scheduled times. Agents don't just act as responsive; instead, they can work in a goal-directed fashion, making them proactive entities and can take the initiative to achieve their goals [1]. Furthermore, agents can adapt to the changes in uncertain situations or a dynamic environment and take the best available actions according to the context [4, 42]. Additionally, agents may use the available knowledge from other agents to take the most appropriate choices and decisions.

From a philosophical point of view, to name a certain software as an *intelligent* is an open argument. But at least minimum requirements should be met in this software to manifest intelligent characteristics and behaviours. This paper defines an *intelligent* software as an ***agent*** that operates autonomously and collaboratively in its environment. Eventually, it can achieve its objectives by reacting to stimuli and communicating with the relevant actors.

3 Related Work

Several works have been carried out to investigate the potential and enabler technologies of digital twins and identify the key challenges that face practitioners during designing and implementing dependable digital twins [52, 18, 34, 46, 24, 20, 14]. Technical difficulties, computational barriers and a shortage of well-founded frameworks and approaches have been reported as part of these challenges.

The complexity level of a particular digital twin can vary from another and depends on the representation scale of the physical system under study. This means that in some digital twins' structures, sub-components, processes, and services of a physical system could be represented as a united independent entity in a digital twin. In contrast, in other structures, the components of the physical system can be grouped into multiple atomic digital twins that lead to having a less failure-prone than the former structures. Still, they have more complex, hierarchical and distributed architectures. As the number of components of a physical system increases, achieving the latter design is more challenging. To clarify this matter, suppose we want to build a digital twin for a specific CPS with multiple components (sensors and actuators). Every part has several internal micro-services that interact with other services. Consequently, all these services should be encapsulated and mirrored into a digital twin. In this scenario, the representation and the integration of physical elements and their micro-services into separate digital twins can be tedious and time-consuming due to the heterogeneity and complexity of the interactions that might occur between these different components and their internal services.

For that reason, it's necessary to use a modular and flexible [38] approach for building modular, re-configurable and scalable digital twins. To realize this objective, some efforts focused on providing ad-hoc solutions for creating digital twins. However, there is a lack of frameworks that offer straightforward, pluggable, reusable, scalable, extendable, and intelligent features that can be used to design and build hierarchical and distributed digital twins and represent heterogeneous components effortlessly [39].

The agent-driven approach has been considered a promising solution to build and deploy a modular, re-configurable, heterogeneous industrial systems [26, 23] and handle the complex communication and interactions of those systems. Leitão and Karnouskos [27] discussed the agent paradigm, its impacts and the main factors that have led the industrial sector to accept it in manufacturing. The conducted review showed promising results for leveraging industrial agents because of their characteristics: intelligence, flexibility, extendability, agility, modularity, responsiveness and robustness that can pave the way for implementing intricate, complex and distributed systems for CPS. Thus, several attempts from the research community and industry [22, 23, 27] were dedicated to using ABM and MAS-based approaches to develop, build and deploy intelligent designs and solutions, including digital twins. However, most studies that utilized agents in the context of digital twins are relatively scarce and new. In the following paragraphs, related works are summarized, listed and reported.

Braglia et al. [5] present an Agent-based simulation model to operate a paper products warehouse. Their implementation integrates the Ultra High-Frequency Radio Frequency Identification (UHF RFID) technology in a digital twin fed with sensor data. It works to optimize the routes, minimize travel distance, and handle possible congestion. The work targets a specific domain. Besides, agent-based modeling was used to model only the digital part in contrast to our approach, which models physical and digital assets as agents but with different functionalities. Real-time communication was not addressed as well in the proposed approach, as the forklifts in the warehouse transmit their relevant position-item data to the simulator every 30 minutes.

In the study by Ambra and MacHaris [2], the paper shows the proof-of-concept of digital twins and how they can be constructed by integrating virtual and physical spaces, which was achieved by feeding the real-time data from the physical environment to the virtual geographic information system (GIS) environment. Agents are utilized to provide the physical twins at ports and terminals such as barges, trucks, trains, vans and parcels with self-awareness capabilities. However, The paper investigated the potential of digital twins, particularly in the synchromodal transportation domain. The introduced approach focuses on just representing physical elements as agents in a virtual environment. Also, none of the proposed solutions tackled the limitation in communicating information from the virtual system back to the physical system.

Implementing a digital twin in smart cities was discussed in the work of Clemen et al. [10], which utilizes IoT for incorporating real-time data into agent-based modeling simulation (ABMs). The presented work implemented a digital twin using the MARS framework for large-scale MAS and presented the entire process incorporating the model description and the real-time data retrieved from IoT sensors. An experimental setup that uses an existing simulation model of Hamburg's traffic system was integrated with the real-time sensor network. Nonetheless, the approach is built and designed using a specific MAS called MARS, tailored for specific domains such as social ecology or simulations related to urban living and transportation. In our proposed framework, the agent-oriented programming and the agent platform are rather technical and not conceptual specifications. Also, the approach covered only one level of complexity, suggested in the study, the passive digital twin instance (DTI), where a software agent is connected to a physical sensor and incorporates state changes, so the connection is unidirectional, not bidirectional, as we proposed. Due to their vital importance to humans, agriculture and farming are also considered in the domain of digital twins. Advanced technology can be utilized to improve farming methods.

The work of Skobelev et al. [48] discusses developing a digital twin for a plant by utilizing multi-agent technology and a knowledge base on macro stages, which allows monitoring and controlling the plant in many stages (plant development, vegetation quality, timing of following stages, and recalculation of forecast). Yet, the digital twin implementation targets a very specific domain (wheat plant farming). In such a domain, a digital twin is limited to monitoring, simulating and predicting the physical environment, and it cannot directly control or affect the physical asset. Also, the paper lacks a clear, well-defined framework that could be re-used or extended in other domains; challenges such as real-time communication, efficient data gathering and coupling methods were not discussed clearly enough.

Zheng et al. [54] proposed a method for modeling a digital twin based on MAS architecture. The study focuses on quality control in manufacturing, providing relevant information, and analyzing product quality during the manufacturing processes. The basis of the DT model is five parts, the physical entities, virtual models, DT data, services, and connections between the components. The approach focused on quality control during the manufacturing phase. The architecture of this digital twin has only one MAS implementation, representing the physical production system. Thus, there is no separation of concerns between agents of the physical asset and agents of the digital asset.

The proposal of Latsou et al. [25] suggests integrating a digital twin into a cyber-physical manufacturing system (CPMS) with the support of using RFID technology to enhance the traceability and trackability of intricate manufacturing processes. The objective is to enhance the manufacturing system's efficiency by providing real-time data and analytics to enable manufacturers to identify and optimize potential issues. The study also considers interactions within a single manufacturing system and multiple sites across a supply chain. The proposed architecture focuses on using MAS technology as a service but not modeling the entire system (physical and digital assets).

Using MAS technology and digital twin in the energy management domain was discussed by Massel and Massel [30]. The authors introduced the idea

Table 1 Comparison table of the related work. Legends, agent-based digital twin: AB-DT, bi-directional integration: ⇌, uni-directional integration: ⇀.

Paper	Architecture	Physical Digital Assets Integration	Real-time support	Ontology	Application Domain
[5]	✗	Physical⇀Digital	Partially	✗	Supply Chain and Logistics
[2]	Domain-specific	Physical⇀Digital	✓	✗	Transport and Logistics
[10]	Integration architecture	Physical⇀Digital	✓	✗	Smart Cities
[48]	Domain-specific	Physical⇀Digital	Partially	Domain-specific	Agriculture
[54]	Domain-specific	Physical⇀Digital	✓	Domain-specific	Manufacturing
[25]	Domain-specific	Physical⇀Digital	✓	✗	Manufacturing and Supply Chain
[30]	Integration architecture	Physical⇀Digital	✗	Domain-specific	Energy management
This paper	Flexible AB-DT architecture	Physical⇌Digital	✓	AB-DT Upper Ontology	General-purpose

of "communicating a material (CM) concept", which refers to an intelligent product with unique composition and decomposition abilities. The recursive architecture of MAS was used, where intelligence distribution of agents on different levels for CM energy management was discussed. However, the current work does not provide a digital twin implementation. It states that a recursive digital twin to manage energy and monitoring data of sensor nodes will be addressed in the future.

In the domain of agents, ontology has been widely utilized [28, 43] to express the terms, concepts and relationships for the domain application. The feasibility of integrating two multi-agent systems, PEDA [12], and COMMAS [31], was studied by Catterson et al. [6]. The mappings between the two ontologies had to be defined for PEDA agents to be able to communicate efficiently with COMMAS agents. This task was not easy and straightforward to do, as was emphasized by the authors.

However, all previous works that focused on digital twin modeling introduced digital twin implementations where just the physical asset elements are modeled and represented in an agent-based environment. This limits the potential of digital twins to interact and influence the environment.

Overall, the major advantages of our framework over the other implementations: (1) proposed architectures in previous works are more focused on very specific domains, while our framework proposed a flexible AB-DT architecture that can be adopted to design and build other agent-based digital twins; (2)

physical components of the physical asset are modeled, programmed and controlled by agents; also, those agents have digital representatives in the virtual asset. Having two layers of agents provides the digital twin with context-aware capabilities where the concerns of agents in the two layers are separated. So, physical agents can operate in the context of the physical system, while digital agents can perform more high-level operations and reasoning. Moreover, (3) providing extendible upper ontology to design agent-based digital twins was also overlooked in previous works. Hence, our framework introduced upper ontology for agent-driven digital twins, which can be extended to different domains. Over and above that, (4) our framework supports real-time communication between physical and digital assets; this feature enables the integration of physical and digital assets and establishes a bidirectional connection where the physical asset sends updates to the digital asset, in accordance with that the digital asset is capable of changing and influencing the state of the physical asset. Table 1 elaborates on our framework's main differences, features and advantages over the previously reported related works.

4 Methodology and The Proposed Framework

This section contains the main content of this paper. In subsection 4.1, we briefly introduce the methodology we follow to conduct this research. Subsection 4.2 spotlights the proposed framework and provides a detailed discussion about the metamodel, the core architecture and the upper ontology. The last subsection 4.3 deliberates on the effective and proper practices of development methodologies to design and build agent-based digital twins for CPS.

4.1 Research Methodology

Researchers and engineers in different information technology fields need a standard, systematic, organized, and well-structured research methodology to help them conduct their research. Design science research methodology (DSRM) provides a set of practices, tactics, principles, and procedures to conduct methodical research for a defined and formulated problem.

For this reason, we are adopting a DSRM presented in [36] to conduct our research as we follow the guidelines and instruction that governs DSRM. In the following Fig 3, the processes and the road map of the DSRM model are depicted and epitomized. The process starts with identifying the problem and ends with communicating and publishing the results and the artefacts, which eventually should be evaluated rigorously to prove their efficiency, quality, and usability. Additionally, iterations during different phases of the research are quite valuable as new ideas, improvements, and refinements are performed throughout our work.

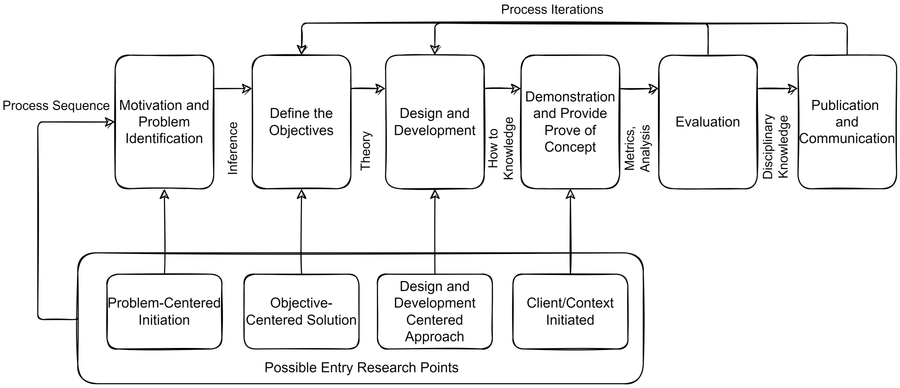

Fig. 3 DSRM Process Model

4.2 MADTwin: Multi-Agent Digital Twin Framework

In this sub-section, a framework named **MADTwin**, which is an acronym for **M**ulti-**A**gent **D**igital **Twin**, for developing agent-based digital twins is introduced. The framework is founded based on the main concepts of the metamodel given in Fig 4 and the generic and flexible high-level architecture highlighted in Fig 5. The upper ontology of the framework given in Fig 6 extends the main concepts of the metamodel and the architecture to enable the modelling of a domain-specific agent-based digital twin implementation.

4.2.1 Metamodel for Agent-based Digital Twin

From a theoretical and abstract perspective, we provide a metamodel that we consider the foundation for designing agent-based digital twins according to our proposed approach. The metamodel is elucidated in Fig 4.

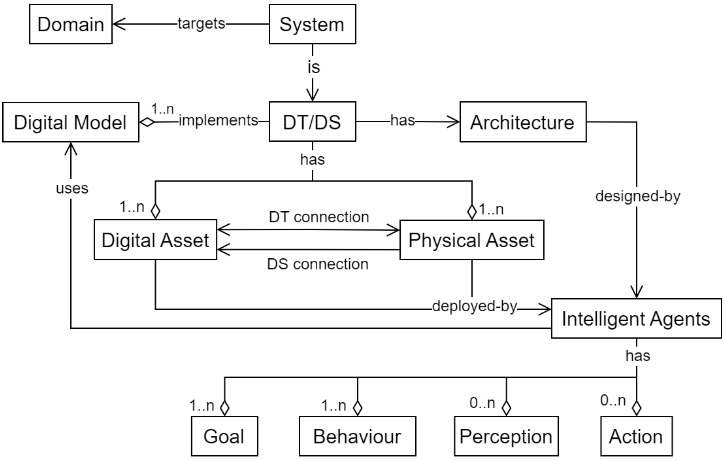

Fig. 4 A Metamodel for Intelligent Agent-based Digital Twin

In this metamodel we impart a conceptual overview of the relations between the `Domain`, adopted `Architecture` powered by `Intelligent Agents` and the two variants `DT/DS` of the implementation, `a Digital Twin (DT)` or a `Digital Shadow (DS)`. Conceptually, a digital twin is a system that targets a specific domain and is constructed according to the requirements of this domain. In theory, once the domain changes, the digital twin implementation and its configurations are modified, adapted, and adjusted according to the new requirements and changes. So, from a conceptual point of view, a digital twin can be constructed with the same architecture and approach. From a technical standpoint, low-level details are the only modifications required when the domain changes. Thus, a flexible architecture is necessary to implement digital twins, provide a systematic approach to fulfil the new requirements and perform the modifications when required.

In the metamodel provided in Fig 4, a digital twin is a system that implements a `Digital Model` of the physical component that contains essential information and configuration about its behaviour, which eventually will be operated in `Physical Asset` and `Digital Asset` based on the applied architecture, which is designed by `Intelligent Agents`. To actualize intelligent characteristics and capabilities, an `Intelligent Agent` considers four primary components, `Actions`, `Perceptions`, `Behaviours`, and `Goals`. Moreover, the system can be a fully functional `DT` or a `DS`, depending on the type of connection between the `Physical Asset` and `Digital Asset`. As depicted in Fig 4, the `Digital Twin` requires a bi-directional channel between the `Physical Asset` and `Digital Asset`. In contrast, a `Digital Shadow` establishes a uni-directional communication from `Physical Asset` to `Digital Asset`.

4.2.2 Architecture for Agent-based Digital Twin

A description of the main components of the generic architecture in Fig 5 is given in the following text. Initially, two pivotal boxes represent `Physical Asset` and `Digital Asset` draw the borders of the two integrated worlds (cyber and physical) that compose the whole digital twin.

Firstly, `the Physical Asset` layer contains `CPS/IoT` components such as sensors and actuators. Agents are deployed on physical components which are enabled by MAS. Thus, physical agents sense and perceive the environment through their physical sensors; also, control, operate and influence the environment through their physical actuators. Every physical software agent assigned for a certain physical part in the `Physical Agents Organization` communicates with its counterpart digital software agent in the `Digital Agents Organization`.

Secondly, the `Digital Asset` layer is also empowered by MAS and contains the digital agents. `Digital Asset` is composed of several other agents; for instance, the `Digital Agents Organization` part comprises only digital agents, who are twins and representatives of the physical agents in the `Physical Agents Organization` of the `Physical Asset` layer. `Digital Asset` layer also includes other abstract agents that have other functionalities which are required to perform different tasks on the digital twin: `Reasoning Agent` is responsible for reasoning and making context-aware decisions in the digital twin; `Simulation Agent` is in charge of conducting simulations about specific context or scenario depending on the data collected from the `Physical Asset` and its agents; `Visualization Agent` is in control of viewing, visualizing and presenting the information collected from the `Physical Asset` and the entire system and which could be accessed by other agents such as external application or human agent; `Learning Agent` could apply learning mechanisms and algorithms to learn from previous experiences to improve the performance of the system and provide intelligent capabilities. The number of agents in the `Digital Asset` depends on the requirements and the essential applications and features the digital twin is built to provide.

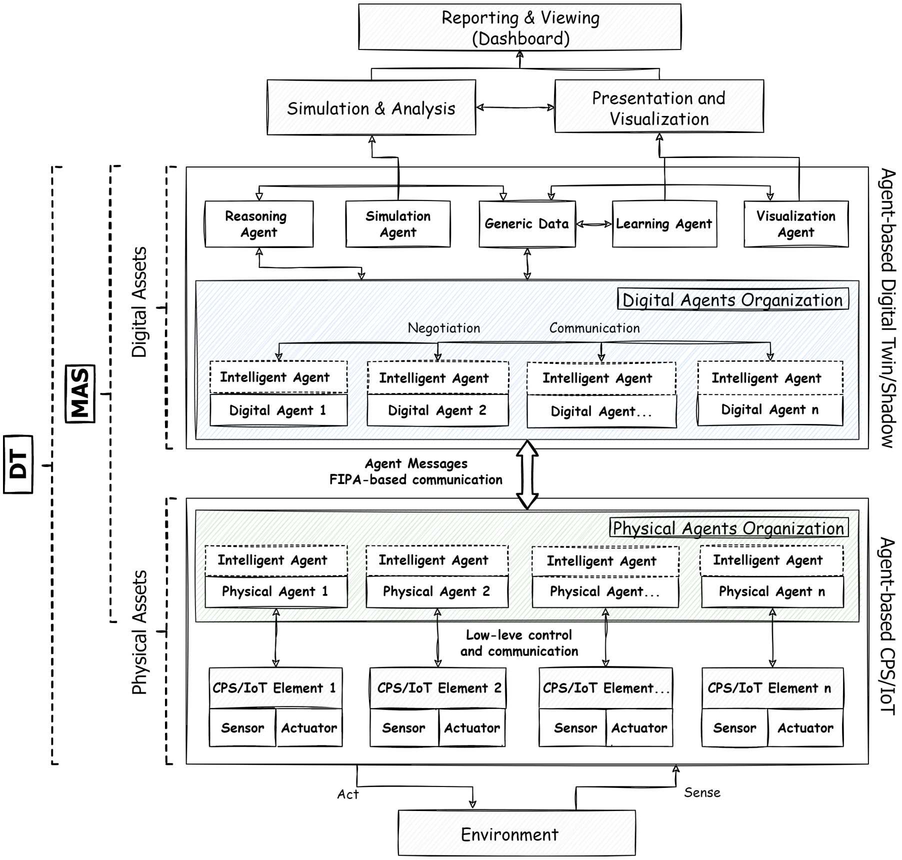

Fig. 5 A High-Level Overview of the Intelligent Agent-based Digital Twin Architecture

Generally, a digital twin can be utilized for multiple and various purposes, including analyzing the behaviour of the system under study, predicting or optimizing its performance, and reporting and visualizing the physical system's state. Thus, to provide these services in the agent-based digital twin, using other technologies is inevitable to cope with specific challenges and enrich the users' experience of the digital twin.

4.2.3 Ontology for Agent-based Digital Twin

An ontology outlines the fundamental concepts and relationships that exemplify a particular domain's vocabulary, terms and rules. Ontology merges concepts and relationships to construct extensions for these vocabularies [16] and build more sophisticated domain-specific designs. An ontology represents concepts and their relations explicitly. Criteria for designing ontology for numerous domains have been introduced [16]. Among the requirements of ontology design are: clarity of ontology, so terms and concepts intended to be communicated can be easily comprehended; coherence and logical consistency of its terms and concepts; extend-ability of the shared vocabularies and concepts; minimal encoding and clear representation [16]. Ontology can capture the knowledge of a specific domain and represent this knowledge in terms, concepts, relations and natural language that are easy to understand. Also, ontology has been utilized to share and reuse knowledge across various disciplines and interrelated domains.

If the deployment of MAS technology becomes widespread in a specific domain, then the demand for interoperability between systems from different vendors and providers will emerge. In this case, the cost of creating several mappings between different deployments may not be feasible and rewarding. As suggested in [6], one option would be to develop and propose an upper ontology for applications in this domain to simplify the integration between different implementations and lessen the burden of starting from scratch to define the main concepts of this domain.

Following this concept, we introduce an upper ontology for the digital twin domain deployed based on the agent paradigm. The presented ontology allows for representing digital and physical assets in the form of digital and physical agents, respectively, using the same terms and vocabularies given in the ontology model. Fig 6 shows the ontology for both the `Digital Asset` and the `Physical Asset`. From a technical point of view and as depicted in the ontology, physical and digital agents are connected and synchronized to each other, forming the twinning concept of the digital twin.

The fundamental and core concepts of the agent-based digital twin domain would be contained in the upper ontology. The ontology would ensure that different targeted domains (i.e., application fields) which are deployed with an agent-based digital twin framework would employ the exact fundamental representation and the standard concepts and terms such as "Digital Agent", "Physical Agent", "Digital Agent Organization", and "Physical-Agent Organization", and how they are related. However, the upper ontology is not sufficiently detailed for specific domain applications. Thus, the ontology should be extended to a specific application domain where it represents the knowledge in that target domain clearly and precisely.

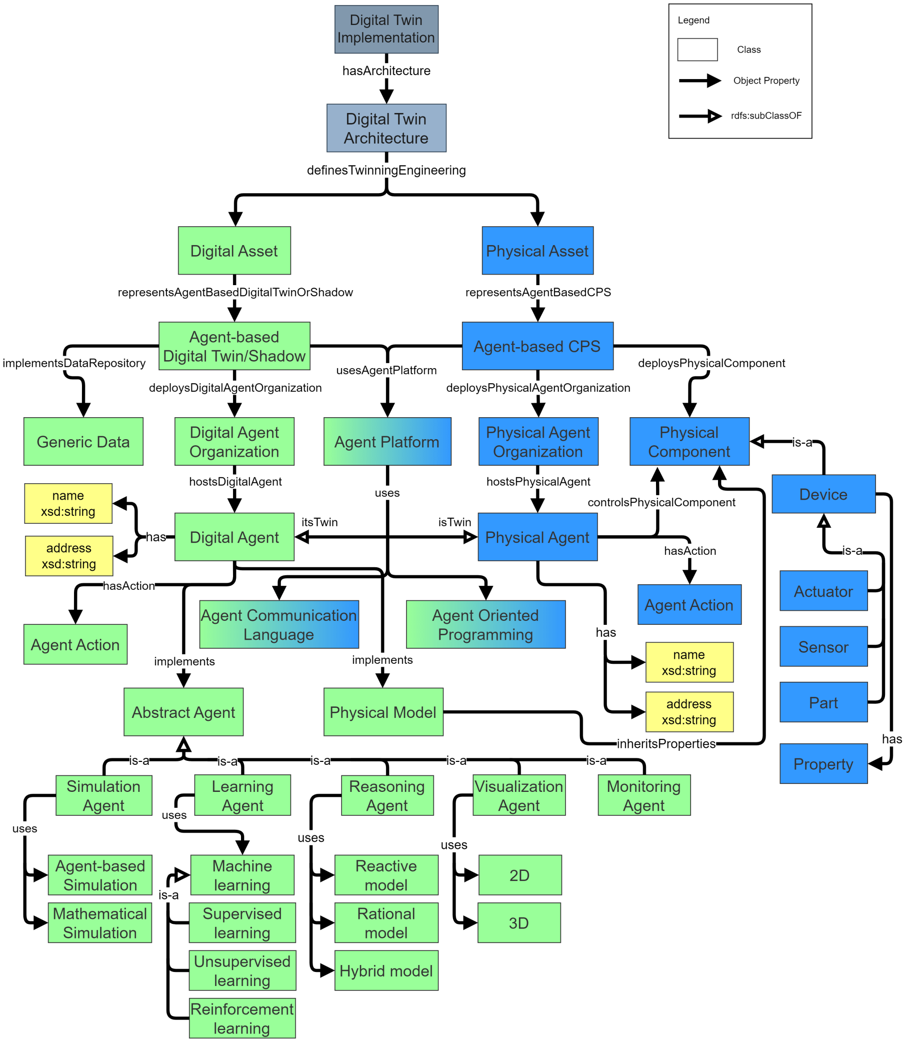

Fig. 6 An Ontology for Agent-based Digital Twin

Following this approach, the upper ontology defines the high-level terminology about both physical and digital agents that compose `Digital Asset` and the `Physical Asset` of the digital twin. The lower (domain-specific) ontology is defined by the domain experts. It covers the detailed terms required by agents within the application field and the area of interest, such as a particular measurement, agent communication standard, sensor type, actuator action, hardware model, or data processing type that is only of interest to agents within that application. Basically, modeling of upper ontology has been done with Protégé ontology editor, where definitions of concepts are provided, and reasoning about the relationships is validated with this tool.

To sum things up, the upper ontology of the agent-based digital twin could be used as a foundation and a starting point for modellers and developers [32] to design and create application domain ontology for different agent-based digital twin implementations, which can reduce the ambiguity of using the domain terms and the complexity of mapping between agents of the physical and digital assets. Inheriting and extending the fundamental concepts from the upper ontology will ensure that many different agent-based digital twin deployments share the same high-level terms and vocabularies. This leads to achieving interoperability between different agent-based digital twin implementations. The case study section provides a concrete example of using and extending the introduced upper ontology.

4.3 Development and Deployment Processes

Large-scale software development comprises several demands and requirements, such as multi-disciplinary teams of developers who are experts on different domains of interest [37].

A digital twin is a rather complex system [18, 46], and it can be deployed on a large-scale based on the domain and the application area. Anyhow, digital twin development comprises two essential phases: designing and building the `Physical Asset` and developing and integrating the `Digital Asset` with the physical one. Depending on the development process of the physical system and the digital system, there are three patterns of digital twin development for the physical and the digital asset that we have observed: 1) the physical system already exists, and digitization of that system is the primary requirement; 2) the physical system does not exist, both a `Physical Asset` and a `Digital Asset` have to be designed from scratch, built, and integrated; 3) the `Physical Asset` is designed and deployed in a simulation. So, `Digital Asset` will be designed and developed first, which will be integrated with the simulation. Then the physical system will be built on reality, and the migration from simulation integration to physical integration should be performed.

The previous patterns impose on developers the need to consider a development model to realize the digital twin and fully integrate the physical and digital assets.

4.3.1 Parallel Development

Usually, teams of developers use a parallel development model to build and develop complex and large-scale software systems. Even though this development model is an essential characteristic in several large-scale and complex systems, following this model gives rise to a set of problems and challenges that should be considered [37]. However, developing a digital twin can be realized with this model, which means developing both the `Physical Asset` and the `Digital Asset` in parallel. This development model has multiple advantages, such as the short time span of delivery of the final product. Yet, challenges and issues may arise due to a lack of communication and understanding between different teams responsible for realizing the physical and digital assets.

4.3.2 Sequential Development

In several cases of systems development, it's crucial to complete a particular phase before starting the next stage of development. A sequential development model divides the system development process into independent phases meant to be finished in order, i.e., the next step shouldn't start before the precedent one is completed [33]. In the context of digital twin development, developing physical and digital assets is under the hood of the digital twin life cycle. Hence, following the sequential development model could be more effective in some scenarios as the digital twin development starts from the `Physical Asset` development and ends with the `Digital Asset` development and integration. According to our observations, following the sequential development model could give developers an explicit and vivid picture of the `Digital Asset` development phase as the `Physical Asset` is already designed and built, and the requirements are concretely defined.

5 Case Study: Prototyping an Agent-based Digital Twin for Smart Warehouse

5.1 Motivation

The internet has drastically changed the shopping concept as the retail industry adopted online shopping methods. This new way of shopping imposes enormous pressure on logistic companies and retailers to provide their goods to their customers without delays and mistakes. Thus, managing warehouses where the goods are organized, packed, and shipped to the relevant address needs to be operated with more automated methods, such as using mobile robots and intelligent systems to avoid human mistakes and decrease the load on workers. In this regard, deploying mobile robots in warehouses [3] requires an ecosystem with all its components and sub-services and which are managed and orchestrated by an intelligent system. In this system, the user can monitor and analyze the behaviour and performance of the warehouse and the robots. Also, the features of performing simulations based on particular scenarios or settings can help to obtain information and insights regarding the system and how it can be improved. Following this goal, digital twin technology is a good fit solution to provide operational and management tasks for such systems. To this end, our case study mainly focuses on constructing an agent-based digital twin supported by multi-robots to realize smart warehouses.

By implementing this case study, we aim to imitate some scenarios and requirements in industrial environments to advance and adjust the implemented system to be a functional prototype for a smart factory warehouse.

5.2 Implementation and Deployment

An initial prototype of the agent-based digital twin has been implemented to show the feasibility and applicability, provide a proof of concept of the proposed framework, and demonstrate its potential and effectiveness. The deployment of the agent-based digital twin utilized the core parts of the framework, which have been described in subsection 4.2.

Depending on the development methodology workflow, the agent-based digital twin deployment phase can be performed sequentially or parallelly. For example, handling and developing the `Digital Asset` layer can be done simultaneously while the CPS `Physical Asset` is deployed in the environment. The two development phases should combine teamwork with a focus on effectiveness and manoeuvrability, which can minimize the gap between the two development phases and utilize the feedback loops more responsively and effectively. This is very crucial as, at a certain point, agents in physical and digital assets should be able to communicate and synchronize with each other.

From a technical viewpoint, deploying the `Digital Asset` and the `Physical Asset` with agents can be done using different agent-based platforms. Whilst the ontology and the agent's communication language should be unified to be interoperable to integrate it with other implementations. In our case, Java Agent Development Environment (JADE)[1] has been utilized to develop both layers. Basically, JADE is a widespread and well-known MAS middle-ware framework. JADE platform offers quite several features for developing distributed multi-agent systems [4]. JADE enabled us to model and develop modular, pluggable and intelligent physical and digital agents by utilizing its functionalities, such as agents behaviour models, FIPA communication standard, and the ability to use JAVA external libraries.

5.2.1 Physical Asset Deployment

As a first step for actualizing our case study, we have designed and built Autonomous Mobile Robots (AMR) prototypes with LEGO technology [44, 53] and integrated them with other technologies and hardware components. We have used the single-board computer represented in a RaspberryPi[2]-BrickPi[3] as the brain and the central processing unit of the robots to provide high processing capabilities.

Smart Factory Warehouse is the idea behind our case study. Therefore, the functional requirements of the digital twin in a warehouse should be determined. In our case, the robots (AMR) in the warehouse should be capable of navigating the environment and travelling to pick up a package from a source point and reach the destination to deliver that package. In addition, to be able to have multiple robots in the environment, every robot should be able to detect obstacles and barriers. Above that, a collision avoidance mechanism should be implemented; in our case, this has been provided by a digital twin

---

[1] https://jade.tilab.com

[2] https://www.raspberrypi.com/

[3] https://www.dexterindustries.com/brickpi

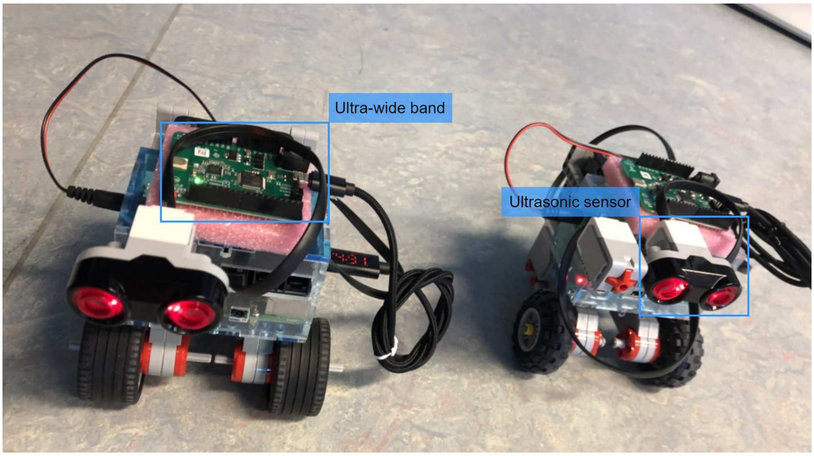

Fig. 7 Overview of AMRs Prototypes Used in the Smart Warehouse Case Study

(i.e., a monitor agent). Other functional requirements, such as time constraints between picking up and dropping off packages or handling missions, could be specified according to the environment and users' preferences.

Initially, the AMRs are designed according to the physical requirements. Each robot comprises two plastic wheels attached to two main separated servo motors and fixed on the same axis, providing the robots with a `Digital Differential` for trajectory-control and movement in the environment. It's essential to determine the robot's chassis characteristics, such as the vehicle width and wheel diameter, which are aggregated and used for adjusting movement and navigation. Robots are also equipped with ultrasonic sensors to detect obstacles and barriers. As a prominent feature, we equipped the AMRs prototypes with the ultra-wideband (UWB) technology provided by [4] for indoor positioning and localization. UWB tags send real-time data regarding the current coordinates of the AMR through the MQTT communication protocol. The abstract UML model of the robot's physical elements is given in Fig 8. Also, the actual physical prototypes are shown in Fig 7.

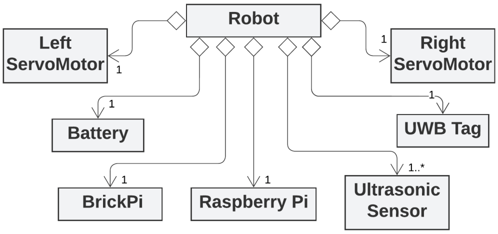

Fig. 8 Model of the Physical System

---

[4] https://www.pozyx.io/

After finishing all the aforementioned stages, the next step is to *agentify* the `Physical Asset`, which means that particular properties of interest in the physical system are abstracted and represented in agents in a MAS platform by programming the agent to control and operate these physical components.

At this stage, we started to utilize the JADE platform to design and program the physical agents. JADE provides a structured framework comprising main constituents; agent language, behaviours and the environment where agents are situated, and they are called containers. Agents are the main abstract elements which have several sub-elements, so-called behaviours. There are multiple types of behaviours the agent can have, and they are categorized into two main categories *SimpleBehaviour* and *CompositeBehaviour* [4].

In addition, the ev3dev[5] Debian Linux-based operating system, which is compatible with multiple platforms like LEGO MINDSTORMS EV3 and the Raspberry Pi-powered BrickPi[6] has been utilized to deploy the agent-based programs on the AMRs.

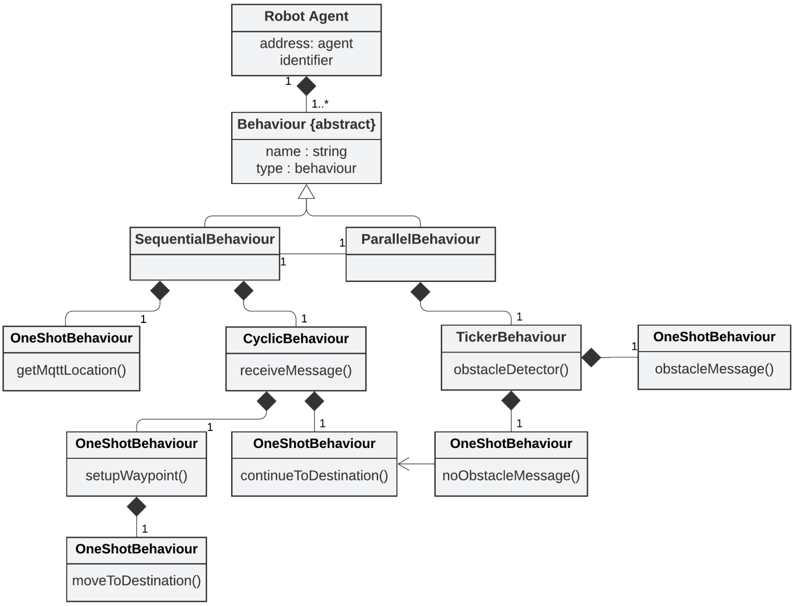

Fig. 9 Model of the Physical Agent

Fig 9 highlights the main behaviours deployed in the physical agent to have autonomous AMR in the warehouse. Each physical agent has its own thread of execution and communicates with other agents using FIPA-compliant messages. Also, physical agents execute every behaviour to achieve a certain goal and their designed objective. A set of behaviours of the physical agent

---

[5] https://www.ev3dev.org

[6] https://www.dexterindustries.com/brickpi

represent the entire functionality of the robot, and the agent could switch between those behaviours according to its operation mode.

For instance, *obstacleDetector()* behaviour is inherited from a scheduling behaviour `TickerBehaviour` provided by JADE platform, which can be executed at selected intervals. Thus, this behaviour waits for a given period of time (specified in its constructor) to execute its internal functions. The *obstacleDetector()* behaviour periodically makes the robot agent check for any obstacle every specific time (e.g., 100 milliseconds) by getting the distance measurements from the ultrasonic sensor. The *obstacleDetector()* is added to a composite behaviour called `ParallelBehaviour` that schedules all its children's behaviours to work in parallel, so, all sub-behaviours can be executed simultaneously.

For instance, in the listing 2, the *obstacleDetector()* is executed every 1 second (1000 milliseconds) and inside the method *onTick()*, the behaviour checks if there is an obstacle has been detected or not and accordingly executes one of the inner `OneShotBehaviour` behaviours which are *obstacleMessage* or *noObstacleMessage* as shown in the code excerpts in the Listing 1.

```
TickerBehaviour obstacleDetector = new TickerBehaviour(this, 1000) {
public void onTick() {
// get the distance measurements from the ultrasonic sensor
    ...
        if (obstacleDistance<35 && !isObstacle)
        {
            isObstacle=true;
            addBehaviour(obstacleMessage);
            removeBehaviour(checkMissionComplete);
        }else if (obstacleDistance>35 && isObstacle){
            isObstacle=false;
            addBehaviour(noObstacleMessage);
        }
    ...
    };
OneShotBehaviour obstacleMessage = new OneShotBehaviour() {
public void action() {
    // stop the robot and send obstacle message to the digital twin so the user get
     notified about the situation
    ...
        }
    };
OneShotBehaviour noObstacleMessage = new OneShotBehaviour() {
    public void action() {
    // send no obstacle message and also continue the current mission if it's finished or
     start with a new mission
    ...
        }
    };
```

Listing 1 Internal Behaviour of Physical Agent for Detecting Obstacles and Sending Obstacle Messages to digital twin

The behaviour *receiveMessage()* is implemented as in the listing 2 to be used to receive messages from the digital twin agents, and specifically the monitor agent which sends mission messages for different robot agents in the system. The type of the *receiveMessage()* behaviour is a `CyclicBehaviour`.

This behaviour never stops after being added to the behaviour queue, initiated and executed. The advantage of this behaviour is to receive messages once the messages are sent from other agents in the system and continuously check if there any new message has arrived, and by using the *block()* method inside its internal *action()* method we can make the behaviour executed only if a new message has been received according to certain criteria (i.e., FIPA-ACL specifications).

```
CyclicBehaviour receiveMessage = new CyclicBehaviour() {
public void action() {
        ACLMessage receiveMessageFromDT = receive();
        if (receiveMessageFromDT != null) {
        // receive a message with mission details
        ...
        } else {
            System.out.println("Waiting Messages");
            block();
        }
    }
```

Listing 2 Physical Agent Receives Messages From Digital Agent

The type of the behaviour *getMqttLocation()* is a `OneShotBehaviour` behaviour which is basically executed just once. This behaviour gets the current coordinate of the physical agent (i.e., the physical robot) basically by receiving MQTT messages from the UWB tag installed on the robot, so the robot can determine its current location and calculate the distance and the angle (according to kinematics equations of the physical robot) to the target destination point. Both behaviours *getMqttLocation()* and *receiveMessage()* are added to a single composite behaviour `SequentialBehaviour`. Sequential composite behaviour uses a basic sequential scheduling scheme to process its sub-behaviours. Basically, it takes and processes the first child's behaviour. If the first behaviour is done, it moves to the next child's behaviour and continues until no more behaviours are in its scheduling system.

5.2.2 Digital Asset Deployment

At this point, AMRs in a warehouse are operated and controlled by physical agents designed and programmed by the JADE platform. The next step is to represent them in a virtual form (i.e., `Physical Asset` and `Digital Asset`) to construct and constitute the entire digital twin. Therefore, every robot controlled by a physical agent in the `Physical Asset` should be mapped into its digital representative (i.e., digital agent) in the `Digital Asset`. Until now, we have a limited number of AMRs for our warehouse. Accordingly, we have the same number of physical and digital agents as the number of AMRs.

As mentioned in the beforehand section, every AMR with all sub-components (battery, ultrasonic sensor, motors, and UWB tags) has been mapped into a single and individual physical agent, which means we have encompassed all the processes and the features of every robot into just one corresponding agent. Hence, when we want to represent physical agents as digital versions, we must do the same and create just a single digital agent for every physical agent. Following the same concept and the agent-based digital twin architecture given in Fig 5, physical agents are represented as digital agents within the *Digital Asset*. The major distinction between a physical agent and a digital agent is that the latter operates in the cloud (i.e., virtual space), while the former operates on the edge (i.e., physical system). This can make the digital agent more privileged by having access to more resources and computational power. In addition, digital agents can have other functionalities besides the main functionalities of representing the physical agent; for example, a learning process or processing of a vast amount of data can be carried out by a digital agent to decrease the load on the physical agent. This can be achieved properly only if the biological agent is fully synchronized with the digital agent to avoid overdue and asynchronous actions from the physical system side.

Another significant point that is worth discussing is having a physical agent for every property of the physical component (i.e., AMR). This, of course, will create a huge society of physical and digital agents. If the number of physical agents increases, then the number of digital agents will increase as well, which can lead to overhead and upsurge in communication caused by having many twins (i.e., agents). Managing a vast number of agents could be a really big challenge. In contrast, capabilities such as multi-scaling, adaptivity, and resilience could be achieved in digital twins. For instance, if a particular sensor that is represented as a physical agent has failed and broken down, still in such a case, the system can continue working as the other sensors are operational. The same is in the case of the agents, if one agent unexpectedly stopped working, other agents could continue without any problem. This concept will not be valid if we represent all the sensors and actuators of the AMR in a single agent. The latter can communicate and perform better, while the former is more fault-tolerant and flexible if a new sensor is added, we don't have to shut down the whole system, and we have just to add this agent to the system.

Fig 10 clearly describes the digital twin deployment. Moreover, it shows the `Digital Twin Life-Cycle` that reflects the stages of realizing a fully-fledged system with both assets (*Physical and Digital*), and how those assets can be tested, maintained and evolve after implementation and operation, for instance, by upgrading some functions and components.

5.2.3 Messaging and Communication Protocols

The primary means of interaction and communication between agents is through agent messages. The FIPA ACL[7] message standard is one of the most popular agent communication languages. The ability to use several content languages and the management of conversations using established interaction protocols are the main aspects of FIPA ACL. A message's structure consists of a number of key values expressed in FIPA-ACL (e.g., performative, sender, receiver, content, ontology, etc.) [4]. The communicative act, or CA, is an

---

[7] http://www.fipa.org/repository/aclspecs.html

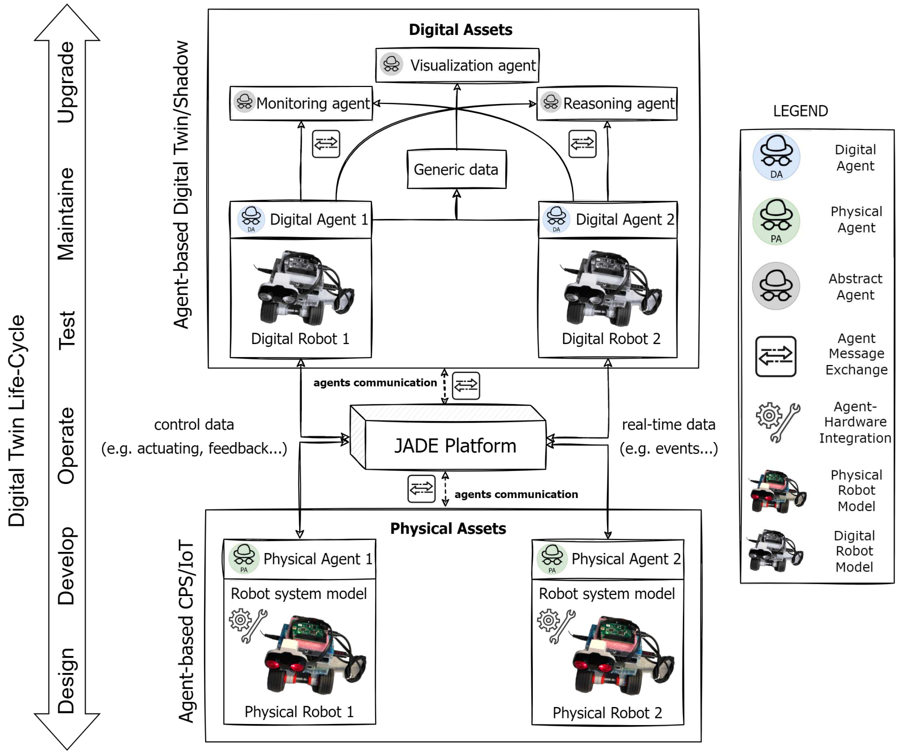

Fig. 10 digital twin Deployment and Development Life-Cycle

action that defines the terms of communication between sender and receiver agents (e.g., request, inform, propose, cancel, etc.), which are defined by the FIPA-ACL [4].

JADE had adopted and maintained the FIPA-ACL as the primary agent communication language. Since the agent-based digital twin deployment has been realized with the JADE platform. Consequently, physical agents and digital agents communicate through the FIPA-ACL agent communication language. Fig 11 shows the sniffer of the JADE platform that captures the messages and the interactions between specified agents in the agent-based digital twin. The contents of the messages, if they are not encrypted, could be displayed from the interface.

Physical agents in the agent-based digital twin communicate with the physical components through a physical interface and the provided hardware API. On the other hand, the localization and positioning of the UWB tags of the AMR communicate with physical agents through the MQTT communication protocol. The UWB tags installed in the AMR subscribe to the MQTT message broker. Accordingly, the updates of coordinates for every specific UWB tag are sent from the master tag of the positioning system and are published through determined topics.

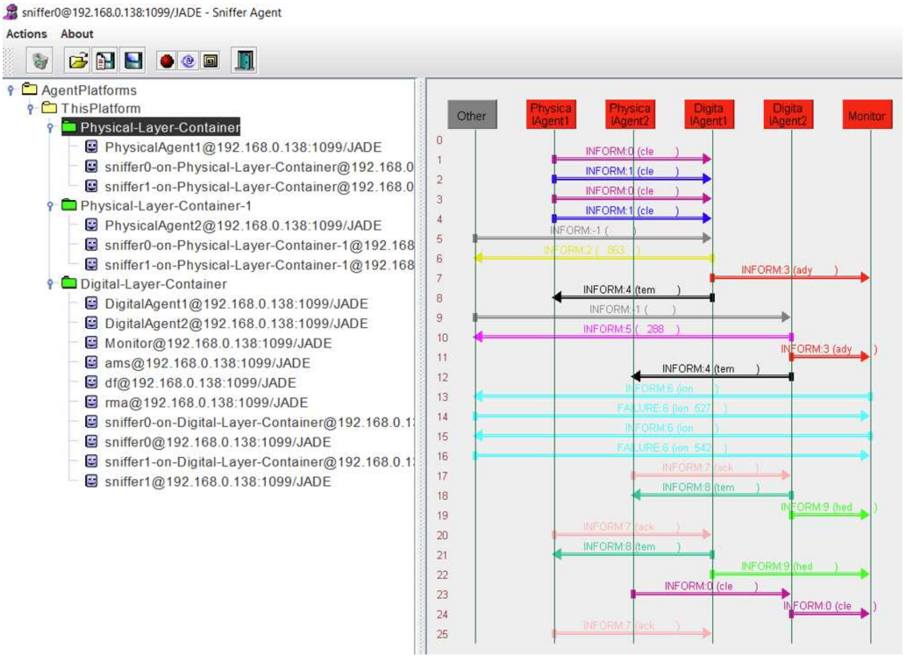

Fig. 11 JADE Interface that Shows Interactions between Agents of the digital twin

5.2.4 Ontology for Multi-Robot Smart Warehouse

In the context of the agent-based digital twin for a multi-robot warehouse, ontology would refer to the set of concepts, terms, and relationships used to describe the elements and behaviours of agents for robots, besides, including concepts related to their capabilities, goals, actions and decision-making processes in the warehouse domain. It would also include information regarding twinning and integrating physical and virtual components, data sources and behaviour models.

Domain ontology would encompass the relationships between upper concepts, such as the interactions between different agents or between agents and the physical components of the digital twin. This would enable modelling of complex agents of digital twins and represent their behaviour more realistically and accurately. For instance, actions for both physical and digital agents are described in the domain ontology. This would enable us to design those actions explicitly in agent behaviours to establish a concrete interaction between physical and digital agents.

Using the content languages and ontologies support provided by the JADE platform, a concrete implementation of the domain ontology presented in Fig 12 has been built and developed based on the introduced upper ontology Fig 6.

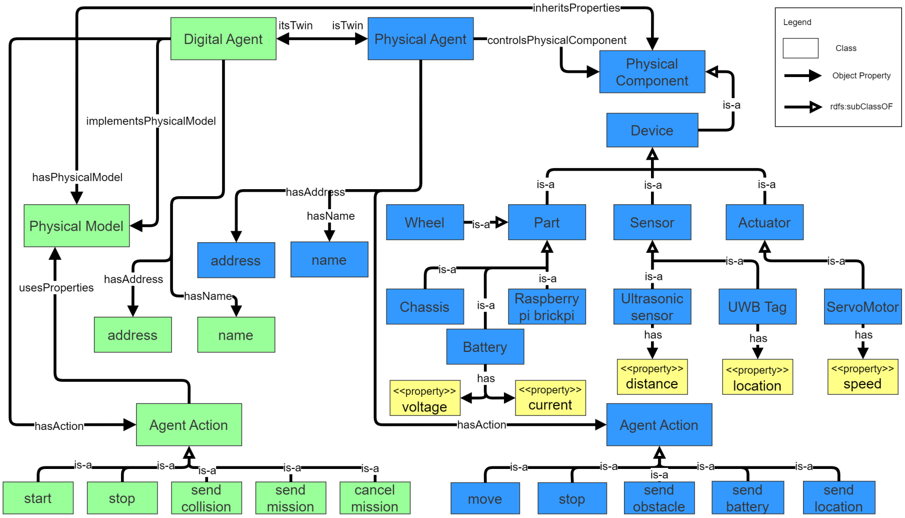

Fig. 12 Partial view of multi-robot warehouse domain ontology

In the example illustrated in Listing 3, agent action `Start` is sent from the digital agent to the physical agent based on the FIPA-ACL agent communication standard. This action message defines the start command to start a new mission.

```
msg = (INFORM
 :sender  ( agent-identifier :name PhysicalAgent_1@Digital-Twin-Platform  :addresses
     (sequence http://DESKTOP-HMMU8GR:7778/acc ))
 :receiver  (set ( agent-identifier :name DigitalAgent_1@Digital-Twin-Platform ) )
 :content  "((action (agent-identifier :name DigitalAgent_1@Digital-Twin-Platform) (START
     :SENDER (physicalAgent :name PA1 :aid (agent-identifier :name
     PhysicalAgent_1@Digital-Twin-Platform :addresses (sequence
     http://DESKTOP-HMMU8GR:7778/acc)) :RECEIVER (digitalAgent :name DA1))))"
 :language  fipa-sl  :ontology  Warehouse-Multi-Robot-Ontology )
```

Listing 3 Agent action (Start) sent from the digital agent to physical agents to start a mission

In another example, an ontological term `IsTwin` identifies that a specific digital agent is a twin of a particular physical agent which, in order, controls a physical component (robot). This message stores information about physical and digital agents and the status of the twinned physical system. Other agents in the system, such as a reasoning agent, can obtain information about the physical system by receiving this message from the physical agent. In our case, a robot is a physical system equipped with a UWB tag with a unique ID, and this tag pulls its location updates regularly. In addition, the robot has a battery. The message keeps the information about the voltage level and the current location of the UWB tag. Thus, the reasoning agent can obtain all this information from every robot in the system. These data can be used to reason about different situations. Listing 4 illustrates an example of the content of this ontology message.

```
msg = (INFORM
 :sender  ( agent-identifier :name PhysicalAgent_1@Digital-Twin-Platform  :addresses
     (sequence http://DESKTOP-HMMU8GR:7778/acc ))
 :receiver  (set ( agent-identifier :name ReasoningAgent@Digital-Twin-Platform ) )
 :content  "((IS_TWIN (digitalAgent :name DA1) (physicalAgent :name PA1 :aid
     (agent-identifier :name PhysicalAgent_1@Digital-Twin-Platform :addresses (sequence
     http://DESKTOP-HMMU8GR:7778/acc))) (ROBOT :name RobotNo_1 :ipAddress
     \"192.168.0.123\" :uwbID \"2365\" :location \"Point2D.Float[6578.0, 5730.0]\"
     :batteryLevel 10.5)))"
 :language  fipa-sl  :ontology  Warehouse-Multi-Robot-Ontology )
```

Listing 4 IsTwin message contains digital twin and physical system information

The extension of the upper ontology to design agent communication concepts and terms can play a pivotal role in addressing the challenge of achieving interoperability and standardization in the context of digital twin implementations based on agents.

5.3 Experiments and Observations

The graphical interface of the JADE platform in the Fig 11 illustrates the physical and the digital layers of the digital twin implementation that constitute physical and digital agents. In fact, the layers are called containers in the JADE platform. Thus, the *Digital-Layer-Container* consists of digital agents, while the two physical layers; *Physical-Layer-Container* and *Physical-Layer-Container-1* encapsulate the physical agents of the *RobotNo.1* and the *RobotNo.2* respectively. As mentioned before, physical agents are hosted and initialized in separate devices. They communicate with the digital layer container in the JADE by creating a new individual container, which is why we have two physical containers for the two robots. If there are multiple agents for a single robot, the physical container will include all of them.

As an illustration of the settings and the scenario for our experiments, Fig 13 gives an overview of the agent-based digital twin functionality and how we use the digital twin to manage, control and distribute the missions among the robots in the warehouse.

5.3.1 Low-level Reasoning

In our running experiment, physical agents operate their corresponding AMRs, representing the `Physical Asset`. Their counterparts, the digital agents, represent the physical agents as a virtual form in the `Digital Asset`. The physical agents communicate with their counterparts or other agents in the `Digital Asset` throughout the FIPA-ACL agent communication standard. The physical agent and digital agent operate in separate machines, and they have different Agent Identifiers (AID), which have a unique address attached to their IP address.

By having agents at different levels of abstraction, we have separated the concerns of physical and digital agents. So, agents have adaptive reasoning and decision-making mechanism. For instance, a physical agent operates in its local context and has overall awareness and control of this space. In contrast, a digital agent can communicate with other agents at the same level and obtain information beyond its operation zone. This provides digital agents with a broader global overview of the system.

This is shown in our experiments in Fig 13 where physical agents can detect obstacles as this is implemented internally inside the physical agent. Thus, AMRs are aware of their operation context. Quite the reverse, the knowledge of the global context is provided by other high-level agents that are located in the `Digital Asset`.

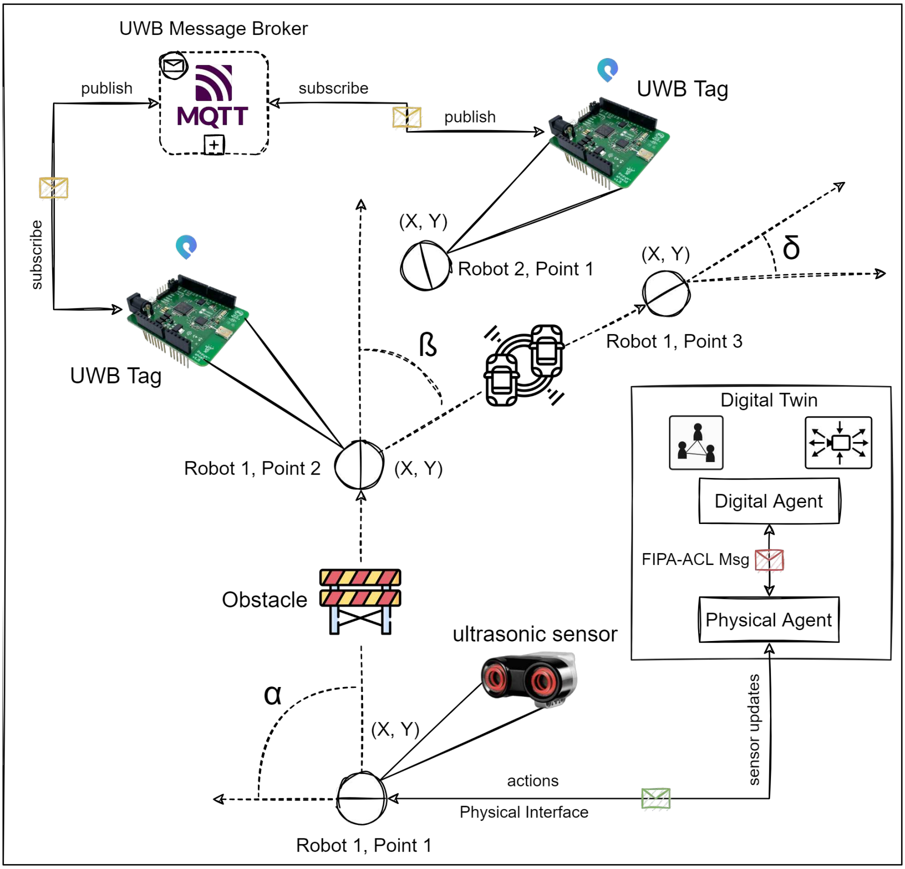

Fig. 13 A Multi-Robot Warehouse Case Study Scenario

5.3.2 High-level Reasoning

Digital agents provide an inclusive overview of the system, which can make it more intelligent as it takes actions based on a broader context. As an example, collision avoidance is orchestrated by the `Monitoring Agent` situated in the `Digital Asset` and collaborating with other digital agents as demonstrated in Fig 10. Thus, the information about the collision cases is broadcast to all physical agents through this agent. Basically, the strategy of separating or distributing reasoning and decision-making process among multiple levels can benefit the system in a way that reduces the load, especially on the edge devices, by conducting more intensive calculations on the cloud side (digital space).

With such an implementation, we could have provided physical agents with autonomous behaviours, enabling them to make self-decisions according to their environment. Additionally, deep reasoning is done in a digital twin by collecting data from physical agents and aggregating these data to obtain a universal view of the entire system. Thus, to a certain degree, we managed to tackle the challenge mentioned in Fig 1 about having intelligent and autonomous capabilities. Ultimately, digital twins are more autonomous with the physical agents' capabilities and with intelligent digital agents that can solve complex situations by exploiting the overall view of the whole system and the interaction with low-level agents.

Based on the observations and results, we have concluded that representing physical elements as agents and designing their digital representatives as agents in the JADE platform provides advantageous features: it separates the concerns of the physical agents and the digital ones (i.e., low-level and high-level reasoning); it couples both agents by taking advantage of the reliable communication mechanism provided by the FIPA-ACL agent language. Accordingly, this realization of such coupling and integration helped to tackle the challenge of integrating physical and digital assets mentioned in Fig 1.

In addition, the challenge regarding providing real-time synchronization between physical and digital agents and physical agents with the hardware has been addressed by utilizing the agent communication language FIPA-ACL and the subscribe-publish network protocol (MQTT), respectively. Anyhow, achieving strict real-time requirements for time-critical systems may require additional layers and special technologies to handle and process data more strictly.

To this end, and by considering and deploying all the requirements and features given in advance, we have made the agent-based digital twin for the factory warehouse more intelligent, safe, and autonomous.

6 Discussion

A digital twin is a tight integration between physical and cyber parts. Implementing a digital twin for CPS poses several challenges that must be overcome as CPSs are naturally decentralized and distributed. These features increase the level of complexity of integrating them into digital twins. Hence, providing a well-structured framework with the conceptual foundation for implementing dependable digital twins is essential. In addition, the framework should provide a flexible, modular, adjustable and re-configurable architecture to build the digital twin and add advanced functionalities to make it more intelligent and robust to alleviate the challenges of the complex systems.

Consequently, in this study, we have proposed a new framework named **MADTwin** for realizing a digital twin for CPS/IoT systems by leveraging the features, potentials, and intelligence capabilities offered by the agent-based approach and the multi-agent systems. Explicitly, we used the JADE programming platform for programming agent-oriented digital twins for multi-robots in a warehouse. We have managed to design and program `Physical Asset` (AMR vehicles) and their representatives in the `Digital Asset` as agents that can communicate and interact with each other.

The **MADTwin** framework focuses on the conceptual part of the agent-based digital twin for CPS/IoT systems, and it adopts a flexible high-level architecture [19] for implementing functional agent-based digital twins. Besides the architecture, **MADTwin** introduces an upper ontology for the agent-based digital twin that serves as a reference for designing and building agent-based digital twin implementations for different target domains and can help to make different implementations more inter-operable. The proposed upper ontology gives an overview of how the main concepts and terms of the agent-based digital twin are structured and related.

The MAS implementation has provided intelligent and autonomous features to the digital twins of the multi-robots, where robots behave autonomously in certain situations. Also, with the use of MAS and exactly JADE platform, the communications between physical and digital agents in physical and digital assets are organized, controlled, and managed adequately and smoothly. In addition, it can help to extend and improve the current digital twin incrementally in a modular way to be more resilient and robust by adding new functionalities to agents, such as reasoning, learning, and simulation.

Overall, the results of this paper imply that using the proposed framework supported by agents is practical and applicable to designing and building digital twins for complex CPS. Yet, some limitations could not be tackled by just utilising the agent approach, and there is a need to merge the current framework with other technologies for handling very specific challenges. Thus, several improvements and features can be added to the framework to make it more functional, intelligent and robust for deploying digital twins.

7 Conclusion & Ongoing Work

In conclusion, this paper proves the feasibility and applicability of the proposed framework. It shows how the main components, including the architecture and the ontology, could be utilized to build a digital twin.

The main contributions of this paper can be abridged in the following. Firstly, it presented **MADTwin** framework for designing agent-based digital twins for complex and distributed CPS that is supported by a flexible and general-purpose architecture. Secondly, it defines an upper ontology that can be modified and expanded to guide practitioners in implementing domain-specific intelligent agent-based digital twin solutions. It can also serve as a foundation for achieving interoperability between different implementations. The third contribution focuses on achieving integration and coupling between physical and digital assets and establishing reliable communication through the use of two types of communication: IoT sensing technology represented in MQTT for handling data streams and the FIPA-ACL communication standard for managing agent-agent control and update interactions. The fourth contribution suggests some development methodologies and practices that can be followed while designing an agent-based digital twin for CPS. Lastly, it elaborates on the advantages and disadvantages of the proposed framework to identify the gaps and the areas for future research.

However, we suggest focusing on some points in order to improve the functionality, performance, useability and interoperability of our framework. For example, we intend to apply intelligent reasoning methods and algorithms, such as considering a fuzzy logic or BDI agent model [8] to reason about complex situations. This will enable physical and digital agents to make better decisions, deal with deviations and uncertainties in physical components, and generally improve the digital twin's performance and functionality.

We also want to address the challenges of efficiently gathering and handling high-frequency data from physical components. Jittering, delays, and inaccuracies in data may cause undesirable results and damages. To enhance the performance of our framework, we will provide dependable mechanisms to process and extract the information from the streams [11] to detect anomalies and data inaccuracy.

Most agent-based and multi-agent system programming languages and frameworks are not trivial to use and deploy. They need much effort, time, and programming skills to utilize. In this regard, and to increase the usability level of our platform, we plan to address the challenge of implementing digital twins with less time and effort. Thus, we intend to consider using a low-code development approach and model-driven engineering [9, 21, 29, 50] concepts and techniques to add another layer of abstraction for our **MADTwin** framework, which could simplify and speed up the process of building digital twin by having multi-purpose physical and digital agent models that can be altered and used for new digital twin implementations.

One of our priorities is building an agent-based digital twin that can be used in manufacturing and adopt a pathway towards Industrie 4.0 [38]. Thus, we aim to integrate the I4.0 standards such as Reference Architectural Model Industrie 4.0 (RAMI 4.0) [17], OPC Unified Architecture (UA) [13] and Asset Administration Shell (ASS) [35] in our framework to achieve interoperability and easy integration with industrial solutions.

Declarations

- Conflict of interest: The authors have no conflicts of interest to declare that are relevant to the content of this article.

- Data Availability: Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

[1] Rashi Agarwal, Supriya Khaitan, and Shashank Sahu. Intelligent agents. *Distributed Artificial Intelligence: A Modern Approach*, page 19, 2020.

[2] Tomas Ambra and Cathy MacHaris. Agent-based digital twins (abm-dt) in synchromodal transport and logistics: The fusion of virtual and pysical spaces. *Proceedings - Winter Simulation Conference*, pages 159–169, 12 2020. ISSN 08917736. doi: 10.1109/WSC48552.2020.9383955.

[3] Kaveh Azadeh, René De Koster, and Debjit Roy. Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917–945, 2019.

[4] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, USA, 2007.

[5] M. Braglia, R. Gabbrielli, M. Frosolini, L. Marrazzini, and L. Padellini. Using rfid technology and discrete-events, agent-based simulation tools to build digital-twins of large warehouses. *2019 IEEE International Conference on RFID Technology and Applications, RFID-TA 2019*, pages 464–469, 9 2019. doi: 10.1109/RFID-TA.2019.8892254.

[6] Victoria M Catterson, Euan M Davidson, and Stephen DJ McArthur. Issues in integrating existing multi-agent systems for power engineering applications. In *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pages 6–pp. IEEE, 2005.

[7] Moharram Challenger and Hans Vangheluwe. Towards employing abm and mas integrated with mbse for the lifecycle of scpsos. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, pages 1–7, 2020.

[8] Moharram Challenger, Baris Tekin Tezel, Omer Faruk Alaca, Bedir Tekinerdogan, and Geylani Kardas. Development of semantic web-enabled bdi multi-agent systems using sea_ml: An electronic bartering case study. *Applied Sciences*, 8(5):688, 2018.

[9] Moharram Challenger, Baris Tekin Tezel, Vasco Amaral, Miguel Goulão, and Geylani Kardas. Chapter 8 - agent-based cyber-physical system development with sea_ml++. In *Multi-Paradigm Modelling Approaches*

for Cyber-Physical Systems, pages 195–219. Academic Press, 2021. ISBN 978-0-12-819105-7. doi: https://doi.org/10.1016/B978-0-12-819105-7.00013-1.

[10] Thomas Clemen, Nima Ahmady-Moghaddam, Ulfia A Lenfers, Florian Ocker, Daniel Osterholz, Jonathan Ströbele, and Daniel Glake. Multi-agent systems and digital twins for smarter cities. In *Proceedings of the 2021 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pages 45–55, 2021. doi: 10.1145/3437959.3459254.

[11] Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):1–62, 2012.

[12] Euan M Davidson, Stephen DJ McArthur, James R McDonald, Tom Cumming, and Ian Watt. Applying multi-agent system technology in practice: Automated management and analysis of scada and digital fault recorder data. *IEEE Transactions on Power Systems*, 21(2):559–567, 2006.

[13] OPC Foundation. Opc unified architecture (ua), Apr 2023. URL https://opcfoundation.org/.

[14] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE access*, 8: 108952–108971, 2020.

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

[16] 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.

[17] Martin Hankel and Bosch Rexroth. The reference architectural model industrie 4.0 (rami 4.0). *ZVEI*, 2(2):4–9, 2015.

[18] Weifei Hu, Tongzhou Zhang, Xiaoyu Deng, Zhenyu Liu, and Jianrong Tan. Digital twin: a state-of-the-art review of its enabling technologies, applications and challenges. *Journal of Intelligent Manufacturing and Special Equipment*, 2:1–34, 8 2021. ISSN 2633-660X. doi: 10.1108/JIMSE-12-2020-010.

[19] Marah Hussein and Moharram Challenger. Smart cyber-physical system-of-systems using intelligent agents and mas. In *Engineering Multi-Agent Systems. EMAS*, 2022.

[20] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52, 5 2020. ISSN 1755-5817. doi: 10.1016/J.CIRPJ.2020.02.002.

[21] Geylani Kardas, Zekai Demirezen, and Moharram Challenger. Towards a dsml for semantic web enabled multi-agent systems. In *Proceedings of the International Workshop on Formalization of Modeling Languages*, pages 1–5, 2010.

[22] Stamatis Karnouskos and Paulo Leitao. Key contributing factors to the acceptance of agents in industrial environments. *IEEE Transactions on Industrial Informatics*, 13(2):696–703, 2016.

[23] Stamatis Karnouskos, Paulo Leitao, Luis Ribeiro, and Armando Walter Colombo. Industrial agents as a key enabler for realizing industrial cyber-physical systems: Multiagent systems entering industry 4.0. *IEEE Industrial Electronics Magazine*, 14(3):18–32, 2020.

[24] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51:1016–1022, 1 2018. ISSN 2405-8963. doi: 10.1016/J.IFACOL.2018.08.474.

[25] Christina Latsou, Maryam Farsi, John Ahmet Erkoyuncu, and Geoffrey Morris. Digital twin integration in multi-agent cyber physical manufacturing systems. *IFAC-PapersOnLine*, 54(1):811–816, 2021.

[26] Paulo Leitao, Stamatis Karnouskos, Luis Ribeiro, Jay Lee, Thomas Strasser, and Armando W Colombo. Smart agents in industrial cyber–physical systems. *Proceedings of the IEEE*, 104(5):1086–1101, 2016.

[27] Paulo Leitão and Stamatis Karnouskos. A survey on factors that impact industrial agent acceptance. *Industrial Agents: Emerging Applications of Software Agents in Industry*, pages 401–429, 1 2015. doi: 10.1016/B978-0-12-800341-1.00022-X.

[28] Zheng Ma, Mette Jessen Schultz, Kristoffer Christensen, Magnus Værbak, Yves Demazeau, and Bo Nørregaard Jørgensen. The application of ontologies in multi-agent systems in the energy sector: A scoping review. *Energies*, 12(16):3200, 2019.

[29] Hussein Marah, Geylani Kardas, and Moharram Challenger. Model-driven round-trip engineering for tinyos-based wsn applications. *Journal of Computer Languages*, 65:101051, 2021. ISSN 2590-1184. doi: https://doi.org/10.1016/j.cola.2021.101051.

[30] Lyudmila V Massel and Alexey G Massel. Development of digital twins and digital shadows of energy objects and systems using scientific tools for energy research. In *E3S Web of Conferences*, volume 209, page 02019. EDP Sciences, 2020.

[31] Stephen DJ McArthur, Scott M Strachan, and Gordon Jahn. The design of a multi-agent transformer condition monitoring system. *IEEE Transactions on Power Systems*, 19(4):1845–1852, 2004.

[32] Stephen DJ McArthur, Euan M Davidson, Victoria M Catterson, Aris L Dimeas, Nikos D Hatziargyriou, Ferdinanda Ponci, and Toshihisa Funabashi. Multi-agent systems for power engineering applications—part ii: Technologies, standards, and tools for building multi-agent systems. *IEEE Transactions on Power Systems*, 22(4):1753–1759, 2007.

[33] Kjetil Molokken-Ostvold and Magne Jorgensen. A comparison of software project overruns-flexible versus sequential development models. *IEEE Transactions on Software Engineering*, 31(9):754–766, 2005.

[34] 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, 1 2017. ISSN 2351-9789. doi: 10.1016/J.PROMFG.2017.07.198.

[35] Felix Ocker, Chris Urban, Birgit Vogel-Heuser, and Christian Diedrich. Leveraging the asset administration shell for agent-based production systems. *IFAC-PapersOnLine*, 54(1):837–844, 2021.

[36] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[37] Dewayne E Perry, Harvey P Siy, and Lawrence G Votta. Parallel changes in large-scale software development: an observational case study. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 10(3): 308–337, 2001.

[38] Flávia Pires, Ana Cachada, José Barbosa, António Paulo Moreira, and Paulo Leitão. Digital twin in industry 4.0: Technologies, applications and challenges. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 721–726. IEEE, 2019.

[39] Yassine Qamsane, James Moyne, Maxwell Toothman, Ilya Kovalenko, Efe C. Balta, John Faris, Dawn M. Tilbury, and Kira Barton. A methodology to develop and implement digital twin solutions for manufacturing systems. *IEEE Access*, 9:44247–44265, 2021. doi: 10.1109/ACCESS.2021.3065971.

[40] Qinglin Qi, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, and A. Y.C. Nee. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 58:3–21, 1 2021. ISSN 02786125. doi: 10.1016/J.JMSY.2019.10.001.

[41] 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.

[42] J Russell Stuart and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, USA, 2009.

[43] Ahmed R Sadik and Bodo Urban. An ontology-based approach to enable knowledge representation and reasoning in worker–cobot agile manufacturing. *Future Internet*, 9(4):90, 2017.

[44] Ebert Schoofs, Joanna Kisaakye, Burak Karaduman, and Moharram Challenger. Software agent-based multi-robot development: A case study. In *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–8, 2021. doi: 10.1109/MECO52532.2021.9460210.

[45] Melanie Schranz, Gianni A Di Caro, Thomas Schmickl, Wilfried Elmenreich, Farshad Arvin, Ahmet Şekercioğlu, and Micha Sende. Swarm intelligence and cyber-physical systems: concepts, challenges and future trends. *Swarm and Evolutionary Computation*, 60:100762, 2021.

[46] Concetta Semeraro, Mario Lezoche, Hervé Panetto, and Michele Dassisti. Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130:103469, 9 2021. ISSN 0166-3615. doi: 10.1016/J.COMPIND.2021.103469.

[47] Sanjit A Seshia, Shiyan Hu, Wenchao Li, and Qi Zhu. Design automation of cyber-physical systems: Challenges, advances, and opportunities. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9):1421–1434, 2016.

[48] Petr Skobelev, Vladimir Laryukhin, Elena Simonova, Oleg Goryanin, Vladimir Yalovenko, and Olga Yalovenko. Multi-agent approach for developing a digital twin of wheat. In *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 268–273. IEEE, 2020. ISBN 9781728169972. doi: 10.1109/SMARTCOMP50058.2020.00062.

[49] 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.

[50] Michael Vierhauser, Hussein Marah, Antonio Garmendia, Jane Cleland-Huang, and Manuel Wimmer. Towards a model-integrated runtime monitoring infrastructure for cyber-physical systems. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 96–100, 2021. doi: 10.1109/ICSE-NIER52604.2021.00028.

[51] Michael Wooldridge. Intelligent agents. *Multiagent systems: A modern approach to distributed artificial intelligence*, 1:27–73, 1999.

[52] Jiaju Wu, Yonghui Yang, X. U.N. Cheng, Hongfu Zuo, and Zheng Cheng. The development of digital twin technology review. *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, pages 4901–4906, 11 2020. doi: 10.1109/CAC51589.2020.9327756.

[53] Metehan Mustafa Yalcin, Burak Karaduman, Geylani Kardas, and Moharram Challenger. An agent-based cyber-physical production system using lego technology. In *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 521–531. IEEE, 2021.

[54] Xiaochen Zheng, Foivos Psarommatis, Pierluigi Petrali, Claudio Turrin, Jinzhi Lu, and Dimitris Kiritsis. A quality-oriented digital twin modelling method for manufacturing processes based on a multi-agent architecture. *Procedia Manufacturing*, 51:309–315, 2020.