

Article

Ontology-Driven Guidelines for Architecting Digital Twins in Factory Automation Applications

Wael M. Mohammed ^{1,*} , Rodolfo E. Haber ²  and Jose L. Martinez Lastra ¹ 

¹ FAST-Lab, Faculty of Engineering and Natural Sciences, Tampere University, P.O. Box 600, 33014 Tampere, Finland

² Center for Automation and Robotics (CSIC-UPM), 28500 Madrid, Spain

* Correspondence: wael.mohammed@tuni.fi; Tel.: +358-50-447-9119

Abstract: The rapid emerging technologies in various fields permitted the creation of simulation tools. These tools are designed to replicate physical systems in order to provide faster, cheaper and more detailed illustrative analysis of the physical system. In this regard, the concept of digital twins has been introduced to generally define these simulation tools. In fact, and according to the creator of the digital twin term Micheal Grieves, a digital twin is defined as a physical system, a digital replica of the physical system and information flow between the former parts. This definition is simple and generic for describing digital twins and yet, holistic. This broad definition creates a challenge for developers who target the development of such applications. Therefore, this paper presents a paradigm for architecting digital twins for manufacturing processes. The approach is inspired by the definitions of the ISA95 standard and the onion concept of computer applications to create multi-layer and multi-level concepts. Furthermore, and to satisfy the different required features by industries, the approach considers a multi-perspective concept that allows the separation of the digital twin views based on functionality. This paradigm aims at providing a modular, scalable, reusable, interoperable and composable approach for developing digital twins. Then, an implementation of the approach has been introduced using an ontology-based system and the IEC61499 standard. This implementation has been demonstrated on a discrete manufacturing assembly line.

Keywords: digital twins; Industry 4.0; systematic method; simulation; generic approach; industrial application



Citation: Mohammed, W.M.; Haber, R.E.; Martinez Lastra, J.L.

Ontology-Driven Guidelines for Architecting Digital Twins in Factory Automation Applications. *Machines* **2022**, *10*, 861.

<https://doi.org/10.3390/machines10100861>

Academic Editor: Zhuming Bi

Received: 24 August 2022

Accepted: 19 September 2022

Published: 26 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Simulation applications have become necessary tools for engineers and systems' developers to maximize the performance of these systems in recent years. These tools provide low-cost, safe and effective approaches for testing, validating and monitoring various systems [1]. In addition, simulation tools or simulators are intensively employed in training operators and human workers on expensive machines and equipment and critically sensitive processes and operations. Examples of simulators include airplanes, Rubber-tiered Gantry (RTG)s, construction cranes, manufacturing and production systems, robotics, and many others. Historically, the creation of the concept of simulation is unknown. Nonetheless, the concept of simulation has been reported in the 18th century when the midwife Angélique du Coudray created a full-scale replica of an obstetrical mannequin for practicing the parturition process [2]. At that time, the simulation was only performed in the physical world. In recent decades, simulation tools have become more effective as computer and visual technologies continue prospering. For instance, newly developed immersive technologies such as Virtual Reality (VR) and Augmented Reality (AR) have been introduced in many fields to provide a better experience for the user during the training on working closely with robots [3]. This evolution is surely driven by the advancement in the computer science and communication domains.

In the manufacturing and industrial domain, the concept of simulation has been recently introduced in the term Digital Twins (DT) for products and processes. The term Digital Twins was firstly used by Michael Grieves in 2002 [4]. According to J. David et al. in [4], the initial definition of DTs by Michael Grieves in his white paper [5] considers the DT as a virtual representation of a product from a Product Lifecycle Management (PLM) perspective. Furthermore, the authors in [4] indicated that Mr. Grieves illustrated the DT as a three-component system: the product, the virtual representation of the product, and the virtual-physical connection between the product and the virtual replica. This definition has been debated and modified by several scholars according to the publications in [6–9]. For processes, the concept of DT is similar to the product DT but the focus is on the execution and the processes rather than the lifecycle.

The main challenge that faces developers and researchers when studying the concept of DTs is the definition of a digital twin itself. For instance, are DTs considered to be simulation units or monitoring systems? Do DTs require 3D visualization interfaces? Do DTs require human-friendly interactions? These questions and many others formulate the purpose of DTs. In this regard, this paper presents a paradigm for architecting Digital Twins for manufacturing processes. The main focus in this paper is ignoring the precise definition of the DT and considering it as a data consumer and information generator as highlighted in [5,10–12]. After accepting that, this generic definition can be tailored to suit the application. In more detail, the main contributions of this paper include:

- Conducting research for analyzing the techniques and the technologies that are adapted for developing and building digital twins using academic and commercial solutions as information sources.
- Reusing the available standards and techniques in manufacturing, computer science and industrial management domains for reshaping a generic paradigm for DTs.
- Presenting a guideline or paradigm for architecting a digital twin.
- Presenting an implementation example for architecture.

The rest of the article is structured as follows: Section 2 presents a literature review on related topics. More precisely, it is divided into a review of academic results and commercial solutions. In addition, a brief review is focused on DTs in the context of modern aspects such as Industry 4.0. Section 3 presents the generic paradigm and illustrates the facts behind the selected approach. The fourth section introduces the systematic methods of building a DT. Then, Section 5 depicts a use case using an educational and research assembly line. The Section 6 provides a discussion and the authors' view on the concept. Finally, Section 7 concludes the paper and provides the possible future work.

2. Digital Twins: Literature and Commercial Solutions Review

In order to understand the techniques and methods of developing and building digital twins, a review must be conducted. Thankfully, several surveys and systematic reviews have been provided by the scientific community. In addition to scientific outcomes, the commercially available digital twins represent invaluable references for this research. Moreover, the available standards, such as the ISO-23347, include knowledge regarding the development of such systems. Therefore, this section aims at highlighting the methods and techniques that are reported in the literature regarding the creation of digital twins.

2.1. Literature Review on Designing and Developing Digital Twins

The usage of the term “Digital Twin” has been increasing rapidly in the academic community with the manufacturing domain as the major contributor [4,7,13,14]. This leap in the research of DTs is mainly related to the demand by the industrial sector and due to the advances in the Internet and Communication Technologies (ICT) domain [15,16]. In other words, the high computational capabilities, the highly interconnected devices, and the efficient data analysis techniques permit the development and deployment of such intelligent applications.

Therefore, several approaches have been introduced for building DTs. This is clear in the well-presented article by Chiara et al. [17]. As the paper discusses the application of DT in manufacturing, the authors conducted an adequate analysis on the DT for determining the direction of the current research for developing DTs. Among several aspects, the paper focuses on the used software, the nature of the user interface, the communication protocols and the provided services. The majority of the compared DTs share similar features which include data acquisition, 3D modeling, real-time monitoring, and analysis for optimization. This means the development of a DT must commonly satisfy these features which are aligned with the generic definition of DTs. Some examples on that, Zhang et al. [18] present a dynamic scheduling application based on digital twin agents for scheduling and optimizing workshop tasks. Fang et al. in [19] present research on scheduling job shop tasks using digital twins.

Resman et al. in [20] present an approach for developing an architecture model for planning manufacturing processes. In fact, the aforementioned examples employ AI-based techniques for building digital twin tools that allow optimization and scheduling of processes and tasks. This is considered to be one of the most important utilizations of a digital twin.

Looking further for more academic research on building DTs, different methods, techniques, and strategies have been utilized for developing and deploying activities. As an example, Klemenetina et al. in [21] identify four major building blocks for a DT. These blocks include the Physical Entity platform, Data Management platform, Service Platform and Virtual Entity platform. Further, Tekinerdogan and Verdouw in [22] propose a pattern-based digital twin architecture, where each pattern consists of structure to define the components, and the dynamic to define the interactions. Using semantics technology, Li et al. presented an approach for monitoring robot interactions using a semantic-based digital twin [23]. Mattila et al. in [24] presented an approach for building a digital twin using the ROS, Gazebo and Twinbase frameworks. Moreover, as the digital twin topic has attracted many researchers in recent years, several attempts were seen to organize the development of DTs. As an example, Hendrik et al. proposed a taxonomy for building digital twins in [25]. This taxonomy helps the developers to pinpoint their efforts based on the requirements of the digital twin. Another published research suggests a six-layered architecture for developing digital twins [26]. Additionally, Resman et al. presented a five-step approach for developing a data-driven digital twin [27].

These different approaches and techniques are driven by the usage of the digital twin itself. As an example, the term Cognitive Digital Twins (CDT) involves the digital twins with human-like capabilities such as perception and reasoning. In fact, the term “cognitive twin” is relatively new with simultaneous works reported in [28,29]. Abburu et al. [30] presents a conceptual definition and initial implementation in the COGNITWIN software toolbox aiming at cognitive capabilities for optimal operations and maintenance of process equipment and assets. Likewise, Lu et al. [31] presented an approach enriched with augmented semantic capabilities for identifying the dynamics of virtual model evolution, promoting the understanding of interrelationships between virtual models and enhancing the decision-making based on DT in the frame of H2020 project FACTLOG. Recently, Rozanec et al. [32,33] drew attention to the strategy for knowledge graph modeling to construct actionable cognitive twins focused on capturing specific knowledge related to production planning and demand forecasting in a manufacturing plant, whereas Li et al. [34] put the attention on a unified ontology modeling approach based on GOPRR (graph, object, point, property, role, relationship) for co-simulation.

Looking at the number of published papers regarding digital twins, it is evident that DTs are an important technology for both research and commercial use. With the majority of the presented research work tending to address different industries and systems, they still agree on the basics of a digital twin. Mainly, a digital twin is a computer application where it will require data management and proper interfaces. The main difference appears in the usage and employment of the digital twin. As an example, if a digital twin is used

to optimize operations, then it will require an optimization solver, and if a digital twin is designed to simulate the behavior of the physical system, then it will require a modeling feature. However, the majority of the reviewed articles present an ad hoc solution for a specific problem which provides important insights into the specific application. This limitation is key for the objectives of this research.

2.2. Commercial Solutions for Building Digital Twins

The commercial solutions are critically important in such a study as these solutions are featured as realistic, build-to-fit the customer needs, easy to use, and reliable products [6]. In addition, the commercial solutions hint the researchers towards the direction and the trends of the future in terms of research and innovation activities. Therefore, this section highlights some features of commercially available products. Some of these products are marketed as digital twins. While some are marketed as IoT platforms, simulators, suits, studios or even programming languages [35]. Regardless of the naming, all the studied tools can be listed under the generic definition of a digital twin as long as the main concept of having a representation of a physical system that consumes data and generates information stands. In this matter, Table 1 presents some well-known use cases of using the Digital Twin concept in a commercial manner.

Table 1. Commercially available solutions for building DTs.

Vendor	Brief Description
General Electric [36]	General Electric (GE) targets the concept of DT based on the application area. According to [37], Dr. Colin Parris presents the interest of GE in three main areas: assets, network and process. For each area, GE provides a set of applications in order to form and build the digital twin. As an example, for the Assets Digital Twin, GE employs the Assets Performance Management and for the Network digital twin, the applications named ADMS and GIS are employed.
IBM Digital Twin Exchange [38]	IBM provides an open marketplace for asset owners and end users to exchange assets and build digital twins based on shared data. In this regard, the platform is open for anyone to introduce models and data to purchase or use.
PTC Digital Twin [39]	Like IBM, PTC provides the customer with a marketplace that contains more than 130 tools. These tools can be utilized for building DTs. In addition, PTC provides guidance and development kits for building new tools that specifically suit their customers. Regarding the application domain, PTC considers DT to be beneficial in five key sectors. These sectors include: Corporate/CXO, Product Engineering, Sales and Marketing, Manufacturing Operations, and Customer and Technician Services [40].
Microsoft Azure Digital Twin [41]	Azure is a cloud-based platform from Microsoft. It is marketed as an IoT cloud platform for data acquisitions, modeling and estimation. According to the [42,43], the Azure DT allows the customer to build a virtual model based on the IoT data. Afterwards, these models are connected with each other to form the DT.
Ansys Twin Builder [44]	Ansys is a corporation specializing in developing simulation tools for various industries and business sectors. According to [45,46], Ansys Twin Builder exploits predefined modules for building DTs. In addition, it allows integration with third-party applications such as Azure for data collection and modeling purposes.
SAP SE [47]	The solution provided by SAP mainly addresses the resources, products and assets. This solution is marketed as a network of digital twins as described in the white paper [48]. It can provide simulation and real-time estimation of products during the lifecycle. In addition, it provides a modifiable interface for flexible interaction with the end user.
Oracle [49]	Oracle's DT is based on an IoT platform that permits data and information interconnectivity. According to Oracle, the implementation of a DT includes three main pillars. These pillars include Virtual Twins where devices are emulated, Predictive Twins where the data is analyzed, and Twin Projections where whole systems are simulated based on the analysis that is created by the Predictive Twins.
Bosch GMBH [50]	Bosch is specialized in developing a building Digital Twin that consumes IoT data from sensors that are scattered in buildings using Bosch devices. According to Bosch, the Digital Twin is built using the Microsoft Azure IoT platform, and it employs the semantic technology for knowledge reasoning.
Emerson [51]	The Digital Twin provided by Emerson holds features such as an automation system, vendor independence, selective fidelity, open architecture, and cloud ready [52]. In fact, the DT is utilized in safety, training, Knowledge transfer, environmental, regularity, and optimization applications.
ABB [53]	ABB provides several solutions that form digital twins based on the end user needs. In this regard, ABB supports the customers with DTs for the design, system integration, diagnosis, and prediction activities and applications. For example, the virtual commissioning DT for discrete manufacturing, virtual drive tuning DT, and the predictive maintenance DT for vessels.

Table 1. Cont.

Vendor	Brief Description
MATLAB/ Simulink [54]	Even though they are known to be a programming language and/or a tool for mathematicians and engineering, MATLAB and Simulink are capable of providing virtual representation on a physical system for the purpose of testing. This is evident in [55] as there are two main methods reported for building a digital twin. The first method employs a data-driven approach exploiting the deep learning tools in MATLAB. The second method involves a Simulink block network. The latter is known as a physics-based approach.
COMSOL Multiphysics® [56]	Like MATLAB, COMSOL Multiphysics is a multi-disciplinary simulation platform. This platform utilizes mathematical models of the physical systems for the simulation process. The models are flexibly added as an add-on where the user decides what is needed for the application and domain. In addition, the platform provides a very useful interface including visualization of the components based on the simulation parameters.
NVIDIA Omniverse™ Enterprise [57]	Omniverse is a state-of-the-art platform that allows developers and designers to build and simulate systems with a high level of realism [58]. The platform is based on Pixar's Universal Scene Description and it uses the NVIDIA RTX™ technology. The Omniverse platform includes five key elements: Nucleus is the database and collaboration engine, Connect is the data connections plugins engine, Kit is the Software Development Kit (SDK), Simulation is a set of realistic models that allows the user to select or create, and finally, RTX Renderer creates the high realistic simulations for users and developers.
Visual Components [59]	Visual Components offers 3D manufacturing simulation tools for the manufacturing domain. These tools offer several functionalities such as factory layout configuration, process modeling, statistical analysis, shopfloor connectivity, and robotics simulation among other features. In fact, Visual Component also provides compatible adapters that can work with the omniverse of NVIDIA.
Tecnomatix® by Siemens [60]	Tecnomatix is an industry-driven digital twin for manufacturing applications by Siemens. The solution provides features such as virtual commissioning, human-centered design and planning, plant simulation, robotics programming, statistical analysis, planning and processes optimization, assembly simulation, and shopfloor and layout configuration. This digital twin can be customized and scaled thanks to its development environment.

The search for commercial DTs can be considered a challenging task as information is kept from the public. In addition, several vendors and companies market their existing solutions as digital twins which requires extra investigation. However, there are four main groups or categories that appeared to form in terms of commercially available solutions regarding the methods for building and using a digital twin. The first group is the cloud-based platforms, which are also titled IoT platforms. For instance, this group includes Amazon, Google, Microsoft and Oracle cloud systems. This category is based mostly on the data-driven approach where the user uploads data or connects IoT devices that publish the data to the platform. Then, a set of predefined or accustomed functions and operations are introduced in order to model, analyze, and process the data. In addition, this category provides easy-to-use AI solutions for customers. Finally, the end user maps the results with web services for utilization purposes. Overall, this category is highly flexible and generic for end users to utilize in several domains. Hence, several solutions are built using these categories such as ICONICS's building digital twin [61] and DOOSAN's sustainable energy production digital twin [62].

The second group includes vendor-specific digital twins. This category consists of the companies that provide industrial systems, devices, controllers, and hardware that requires tools to simulate the operations. As an example, ABB's RobotStudio, Siemens's TIA Portal, Omron's ACE Software, and the majority of robotics and PLCs vendors. This category is very accurate and detailed in terms of simulating the systems. In addition, it provides the needed visualization for the end user to allow the best interaction. Nonetheless, the DTs in this group may suffer from being very limited only to the vendor's own equipment. In other words, a digital twin in this group will hardly include a model of equipment from another vendor. As an example, the libraries of the TIA portal by Siemens only include Siemens PLC devices or the libraries of RobotStudio by ABB include only ABB robots and controllers. To overcome this limitation, the users may use standards and protocols for communicating between digital twins from different vendors. This practice might be complicated and will increase the time for preparing a functional digital twin of the production system. Thus, it is advisable to reduce multi-vendor setup in the factories.

The third category is the application-specific Digital Twins. This category is focused on a certain application or domain. For instance, SolidWorks, SAP, ICONICS, Visual Components and Tecnomatix by Siemens. The DTs in this category are dedicated to

solving definite problems in a specific domain. As an example, the Visual Components Digital Twin concept includes several simulation applications that can solve different problems in manufacturing such as layout configuration, process modeling and virtual commissioning [63]. Looking at Siemens's Tecnomatix, the digital twin includes very sophisticated models that allows statistical analysis, optimization of processes and virtual commissioning. Adding to that, this digital twin includes a development environment for end users which increases the flexibility and interoperability. In fact, and as this category of digital twins is built to fit specific applications, industrial digital twins can be found mostly in this category. Finally, the fourth group is the generic graphics and simulation group. This group includes solutions such as Blender [64] and Autodesk's Maya [65]. These solutions are mainly used in film making and game making to develop highly realistic scenes and visual effects using its built-in physics engines and modeling tools. This group is not usually exploited in industrial applications as it lacks connectivity to the real world. Nonetheless, it can create a decent simulation of systems, especially in mechanical and civil engineering. Furthermore, and as an advancement, NVIDIA took these concepts to the industrial sector with its digital twin platform Omniverse. As reported in [66], BMW and NVIDIA collaborated to build a highly realistic and accurate digital twin of an automotive factory with the inclusion of human workers. This puts NVIDIA's Omniverse platform as one of the most sophisticated solutions that can be used to develop digital twins.

Overall, there are several business models that are implemented in marketing and operating these digital twins. Most importantly, and it was very noticeable in the commercially available solution, these digital twins use and follow industry standards which allows interoperability and exchangeability of tools and applications. This feature gives the developers room to develop their own applications and thus, exploit these solutions to its maximum.

2.3. Review on Digital Twins Standards

There is no doubt that the concept of digital twins is highly linked with various technologies and standards as illustrated in [67]. According to the authors, the digital twin combines standards from topics such as Physics Entities, Virtual Entities, Data, Connection and Services. Each topic contributes to the concept of digital twins as the use case needs. For instance, if the digital twin connects to real-time data, then it will require following data collection standards that allow such a service. Furthermore, the industry domain may also contribute to or influence the digital twin solution. As an example, developing a digital twin for the marine industry will require the solution to adhere to some standards and concepts of marine life in general. Similarly, the case of manufacturing, mining, military, etc.

This interrelation with other technologies and domains triggered the issue of ISO 23247 Automation systems and integration—Digital twin framework for manufacturing standard [68]. This standard consist of four parts. The first part presents an overview and terminology related to digital twins. The second part describes the reference architecture. The third part presents the relation with the manufacturing elements. The fourth part presents the information exchange concepts and methodologies. Figure 1 depicts the reference model of the digital twin in the ISO 23247. As presented in the figure, the reference model consists of four main entities. The first entity is the user entity which includes all the needed components for interfacing with the digital twin and manufacturing systems. The second entity is the digital twin entity which includes all applications, services, operations, and management components. This is the core entity that builds the models and the simulation acts.

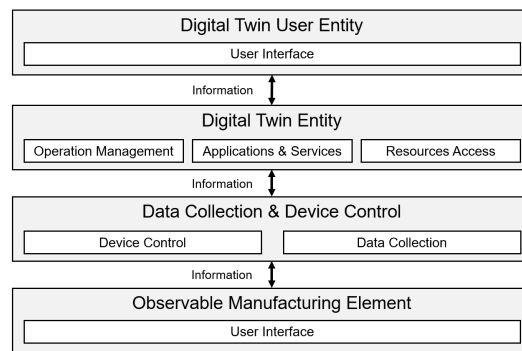


Figure 1. Digital twin framework based on the description in the ISO 23247 standard. [Source: Own figure based on the International Organization for Standardization (ISO) illustration in [68]].

Then, the third entity is the data collection and control entity. This entity is the module where manufacturing elements are connected to collect data from shopfloor devices. Finally, the fourth entity is the observable manufacturing elements entity which includes the physical components. For better understanding, the National Institute of Standards and Technology published use case scenarios of implementing digital twins based on the ISO 23247 [69]. In addition to that, the ISO published a dedicated standard for the visualization elements in automation system under industrial data topic is titled as ISO/TR 24464:2000 Automation systems and integration—Industrial data—Visualization elements of digital twins [70]. This standard focuses only on the visualization elements to be shared between the physical and virtual replicas. ISO is also developing two standards related to digital twin topics. The first one, the ISO/IEC AWI 30172 Digital Twin-Use cases, includes description based on the use case [71] and the second one, the ISO/IEC AWI 30173 Digital twin—Concepts and terminology, on generic concepts and terminology [72].

3. Generic Paradigm for Architecting Digital Twins

Following the review in the previous section, the development of a digital twin depends on several factors including the domain where the DT will be utilized, the understanding of the physical system behavior, the availability of the data, the required level of details in simulation, and the required human interfaces and manner of interactions. Thus, this section presents a generic paradigm of architecting digital twins for manufacturing processes. The main objective of this generic paradigm is to support developers and engineers with adequate guidance for architecting the digital twin regardless of the application or the technologies. In addition, this generic paradigm must be applicable and compatible with different platforms, programming methods and techniques, operating systems, and hardware devices. Consequently, developers will be able to reuse some concepts of the generic paradigm fully or partially based on the application's need.

Generally, the paradigm realizes the digital twin from three points of view. Firstly, the digital twin is a computer application that is deployable on a centralized, decentralized or distributed computing system that is able to interact with other entities and applications using services and proper human interfaces. Secondly, the digital twin is a replica of the physical part of a Cyber-physical System (CPS). Thirdly, the digital twin is a system with one or more perspectives that allow simulating a physical system using virtual replicas. Following these three points of view, the generic paradigm for developing digital twins is a combination of three concepts: *multi-layer*, *multi-level*, and *multi-perspective*.

3.1. Multi-Layer Concept

Developing computer applications may implement well-defined approaches such as Model-View-Control (MVN), Object-Oriented Architecture (OOA), or Multi-Agent System (MAS) among others. These approaches help the developers in architecting the applications based on the problem needs. As an example, the MVC approach helps the developers to separate the application blocks or modules based on the functionality. Meanwhile, OOA is

mainly used in hierarchical architecture where instances are dynamically created and linked with a parent instance. The MAS approach is mainly used in distributed system architecture where agents are permitted to communicate with each other as needed. Another approach for building applications is the onion architecture [73,74]. The onion architecture was developed by Jeffery Palermo [75] which is an implementation of the concept of the clean architecture by Robert C. Martin [76,77]. The onion architecture is a systematic clean method for architecting computer applications by using onion-like layers where the core or the model of the application is in the center and the application interfaces, such as Graphical User Interfaces (GUIs) and Application Programming Interfaces (APIs), at the outside layer of the onion. Some layers can be planned in between the inner and the outer layers if needed. Inspired by the onion architecture, the *multi-layer* concept envisions the digital twin as a computer application with three major layers: *model*, *signals*, and *interfaces*. Figure 2 depicts the multi-layer concept. The *Model* layer represents the core part of the application which includes the logic and behavior of the application. The *Signals* layer includes the data and information management and transformation. Finally, the *Interfaces* layer includes the HMIs and APIs. These interfaces include the human–machine interface and the machine–machine interface. The *signals* layer acts as a linking layer between the interfaces and the core of the digital twin.

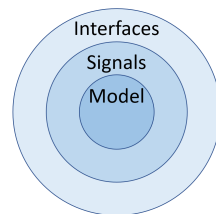


Figure 2. Multi-layer concept following the onion architecture. [Source: Own figure].

3.2. Multi-Level Concept

Following the second consideration of the generic paradigm, a DT must mimic the physical part of a CPS. In industrial applications, Industrial Cyber-physical Systems (ICPS) can be projected on the ISA95 standard (ANSI/ISA–95.00.01–2000) [78] as depicted in Figure 3. As highlighted in the figure, the cyber part is colored in blue and includes the Enterprise Resource Planning (ERP) and Manufacturing Execution Systems (MES) levels. Meanwhile, the physical part is presented in red and includes the production process at level 0 and manipulation and sensing at level 1 of the factory shopfloor. The transition between the cyber and physical parts or worlds occurs at the control and supervisory level (Level 2) of the factory shopfloor. As shown in the figure, the data and information flow is defined following the purpose and features of each level. In this regard, the ERP provides the schedules to the MES level based on the incoming orders and the available resources. At this level, the planning, scheduling and optimization of the production processes are conducted. Then, the MES transfers these schedules into optimized plans and then issues commands to the control and supervisory level to accomplish these plans. The control and supervisory level makes actions accordingly at the manipulation level. These actions affect the production process. As a consequence of that, the sensors in Level 1 detect the changes and send the reactions to the control and supervisory level. The control level provides these reactions as feedback to the MES level. Finally, the MES reports to the ERP the production status.

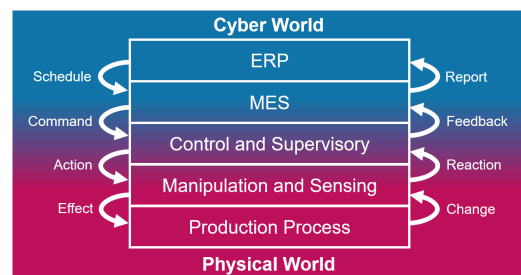


Figure 3. Projection of the CPS concept on the ISA95 levels. [Source: Own figure].

In the context of digital twins for industrial applications, the factory shopfloor levels (Level 0–Level 2) represents the physical part that can be twined. As depicted in Figure 4, and being inspired by the ISA95 standard, the *Multi-level* concept of this generic paradigm includes three levels. *Level 2* is the logic level where all the logic of the production system is included. This level imitates the ISA95-*Level 2* where the control and the supervisory happens. *Level 0* is the level where all the physics of the production system is included. Similar to the ISA95-*Level 0*, this level aims at modeling the production process. Moreover, in addressing multi digital twin situation, *Level 0* is the location where the digital twins can be linked in the physical space. Finally, *Level 1* is the level where the transformation between the logic and the physics occurs. This level represents the manipulation and sensing technologies. Moreover, and as presented in the figure, during production, the MES and the ERP are normally connected to the shopfloor resources. For the digital twin, two main operational modes are presented. These operational modes are activated once the digital twin is developed and deployed. The *On-line* mode when the DT is connected in real-time to the shopfloor resources. In this mode, the digital twin acts as a monitoring system for the physical twin where live streams of data (shown in gray arrows) feed the DT. Meanwhile, the *Off-line* mode is used in simulating the shopfloor resources. In this mode, the DT will be connected to the existing ERP and MES systems for testing and validation of plans and operations.

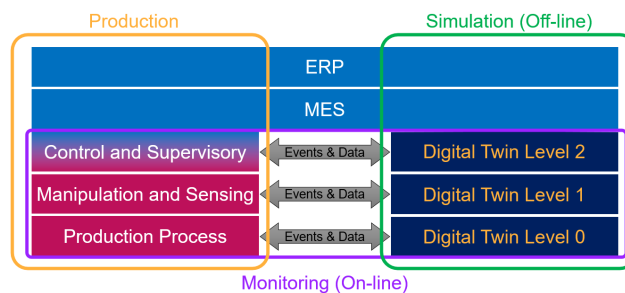


Figure 4. Multi-level concept and the DT operational modes. [Source: Own figure].

By combining the concept of *multi-layer* and the *multi-level*, a digital twin will contain 9 blocks as depicted in Figure 5. For a generic design of a digital twin, the 9-block architecture allows modeling and simulating any production system. As depicted in the figure, the *Model* layer includes all the modeling, logic, behavior and physics simulation components. At level 2, *M2* includes the logic of the control and supervisory. The *M1* block will contain the models of the sensors and the actuator systems. The *M0* will include the models and physics of the process. Then, the *Signals* layer includes all the operations and transformations on the data that will be shared between the *Model* and the *Interfaces* layers. This layer allows the developers of the digital twin to include any needed operations of the data and the information. In this regard, the data and the information can be either generated by the *Model* layer and need to be prepared for the *Interfaces*, or visa-versa, received by the interfaces and need to be delivered to the model layer.

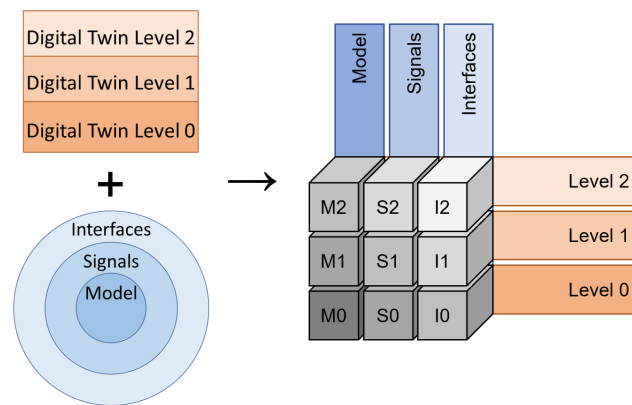


Figure 5. The 9-block concept. [Source: Own figure].

Following that, the *S2* block will contain the transformation components of the control and supervisory. Meanwhile, the *S1* will contain the information transformation components for the actuation and sensing level, and *S0* will include the information transformation components for the process level. Finally, the Interfaces layer is where the communication occurs with human using HMIs and with machines using APIs. The *I2* is dedicated to the interfaces regarding the control level. Similarly, the *I1* is designed for interfaces of the actuation and sensing level, and *I0* for the interfaces of the process level. It is important to mention that the interfaces allow access to inner data and information that is not possible to access in the real system. As an example, in a production system, a motor can be controlled by a drive. This drive includes power electronics that create the sinusoidal signals for the motors. These signals often include the undesired harmonics. Thus, the drives use filters to clean the signals from the unwanted harmonics. In real systems, it is very difficult to study the effect of these harmonics on the other systems in the production systems due to a lack of access. However, in digital twins, this can be replicated and the interfaces layer can allow access to any inner values that the user of the digital twin is interested in.

3.3. Multi-Perspective Concept

Returning to the third point of view for a generic paradigm, a digital twin must provide multiple perspectives to fulfill the different possible applications. In this regard, the *Multi-perspective* concept considers a complete digital twin as concentric cylinders as depicted in Figure 6a. As described before, the levels (in orange) are represented by vertically stacked cylinders and the layers (in blue) are concentric cylinders. To understand the multi-perspective concept, a look from the top of the digital twin paradigm shows different sectors as presented in Figure 6b. Each sector (in green) focuses on a different perspective of the digital twin. As an example, a digital twin can be used for cost estimation and simulation. At the same time, it can be also used for simulating energy consumption or resource utilization. Following the multi-perspective concept, as shown in Figure 6, three perspectives will be needed to address the cost, energy consumption and the resources' utilization use cases. In addition, such a concept allows scaling or extending digital twin usage to address modern or newly developed technologies. As an example, the above-described digital twin can have a new perspective regarding the cognition aspect. This addition will allow the digital twin to be a cognitive digital twin.

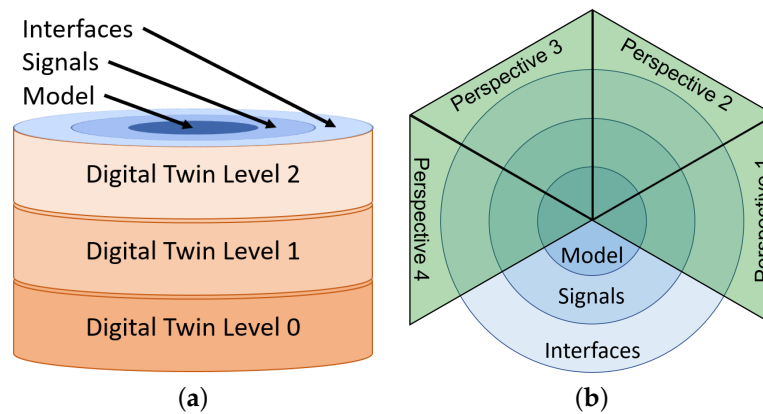


Figure 6. Multi-perspective concept. [Source: Own figure]. (a) Side view ; (b) Top view.

In summary, a generic digital twin will include several perspectives depending on the application and the purpose of the digital twin. Mainly, each perspective includes the 9-block architecture. Figure 7 depicts 3D visualization of the paradigm with different color for each perspective. The main objective of such a paradigm is guaranteeing *modular, scalable, reusable, interoperable* and *composable* digital twin architecture. These qualitative attributes can be mapped to the architecture as follows:

- *Modularity*: is the ability to build the DT using defined and interchangeable modules. These modules contribute to the flexibility of the overall system. In the context of the generic paradigm, the 9-block approach allows the developer to interchange the blocks based on the need of the application.
- *Scalability*: is the ability to grow the system in terms of resources and features. This quality is presented in the possibility of adding several perspectives to the digital twin in order to increase its capabilities.
- *Reusability*: is the ability to reuse legacy or existing assets in building newer versions. In this regard, the 9-block approach allows the user to reuse previously-developed blocks in newer versions of the DT.
- *Interoperability*: is the ability to work and to be compatible with other components or systems regardless of the vendor or the developer. For the 9-block approach, the use of standards and protocols permits such quality where different systems and applications can communicate easily.
- *Composability*: it is the ability to reassemble and reconstruct a system from other systems and components. In the context of this research, the 9-block architecture allows the development of a system of digital twins where the DT Level 0 includes the physics that each DT uses. Then, by connecting each Level 0 using spatial computing concepts, these DTs can form a system of digital twins.

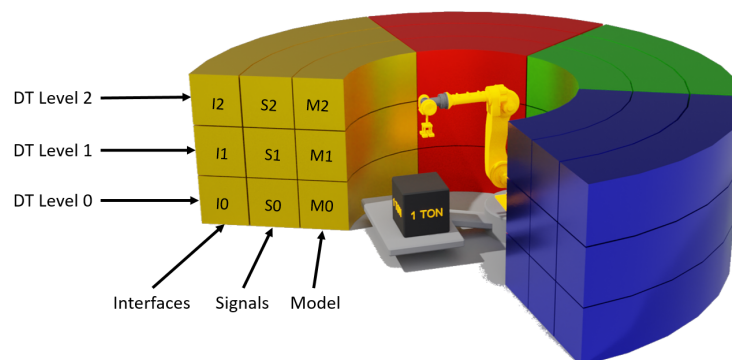


Figure 7. 3D visualization of the generic paradigm for architecting digital twins. [Source: Own figure].

4. The Approach of Architecting the Digital Twin

After establishing the skeleton of the DT using the 9-block concept, developers can start the process of architecting the DT. This process may include the selection of some aspects such as the communication protocol, the shared data, the components and applications, and the development techniques. In the domain of this research, and after defining the 9-block paradigm, architecture mainly refers to two parts. Firstly, the components that construct the digital twin modules, and secondly, the data types and protocols on how these components communicate with each other and with the external applications and services via defined APIs. In this regard, a knowledge-based system can be used to reason the architecture of the DT. As depicted in Figure 8, the reasoning process may require three knowledge bases. Firstly, the *Process description* (green) supports the reasoner with the production resources knowledge. As an example, this ontology may contain descriptions of the production resources such as robots and machinery, product, data and possible events. Secondly, the *Twinning specifications* (red) which include the rules for the reasoning engine, the needed perspectives for the digital twin, and the possible interfaces.

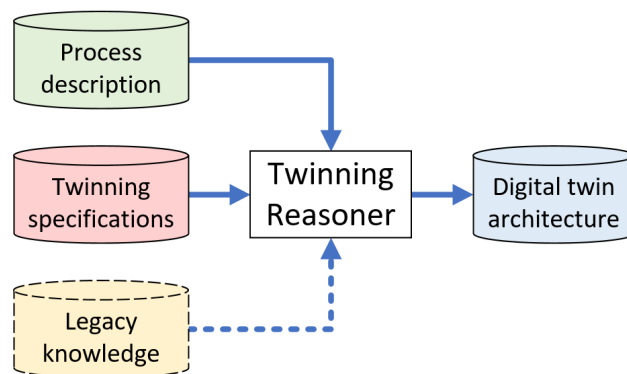


Figure 8. The knowledge-base system for reasoning the digital twin architecture. [Source: Own figure].

Thirdly, the *Legacy knowledge* (yellow) which includes manufacturing vocabulary and descriptions that are required to identify production resources such as robots or conveyor descriptions. In fact, many ontology models are developed and populated for this purpose such as the IOF-BFO developed by the Industrial Ontology Foundry [79–81], the Politecnico di Milano-Production Systems Ontology (P-PSO) and the Manufacturing Systems Ontology (MSO) [82], EU Vocabulary Ontology [83], and the Semantic Computing Research Group (SeCo) ontology [84]. Once these ontology models are utilized, the reasoning engine or as labeled in the figure *Twinning reasoner* will infer the architecture description as an ontology (blue). Figure 9 presets a graph that describes the ontology model of such a knowledge-based approach. This figure shows the relations between the factory and its production resources in the physical world and the digital twins and their components in the digital world. This graph follows the same colors scheme in Figure 8.

More precisely, and starting from the top, the *Production_Control* class represents the MES and the ERP systems that control the production process. As an example, this class can also include the concept of AI-based *Digital Agents*. This *Production_Control* class is linked to the *Production_Process* class via *manages* object property, and to the *Product* class via *monitors* object property. As the figure shows, a *Digital_Twin* simulates the production process or the product. In addition, the *Digital_Twin* virtually replicates the *Factory* and it can have multiple *Perspective* instances linked to it based on the DT specifications. The *Factory* is linked to the *Production_Process* via the property *conducts*. Further, the *Factory* class is linked to the *Product* via the property *produces*. As seen in the figure, the *Product* class and the related object properties are highlighted as this research focuses on the production processes of digital twins. The approach also considers that a factory comprises multiple instances of the *Production_Resource* as shown in Figure 9. The *Production_Resource* can be a robot, machinery or human worker that provides service or conducts an act in the production process. Addi-

tionally, the *Production_Resource* is linked to the *Level* class where the resource is located in terms of manufacturing hierarchy which is needed for the 9-block paradigm. Likewise, a digital twin comprises multiple *Digital_Twin_Resources*. This *Digital_Twin_Resource* class is linked to the *Production_Resource* via *virtually_replicates* object property. In addition, the *Digital_Twin_Resource* may contain several *Digital_Twin_Component*. Each instance of the class *Digital_Twin_Component* is linked to the *Model*, *Signal* and *Interface* classes via *located_at* object property. On the other side, a *Production_Resource* may consume or generates *Data*. Additionally, it can trigger or expect an *Event*. Finally, this approach uses the IEC61499 standard for building digital twin components. In this regard, the *Function_Block* class is linked with *Data* and *Event* via *is_input_to* and *is_output_of* objects properties and is represented as function blocks. As mentioned before, the *Legacy knowledge* (colored in yellow) defines manufacturing vocabulary. This ontology is added to the graph partially for illustration purposes.

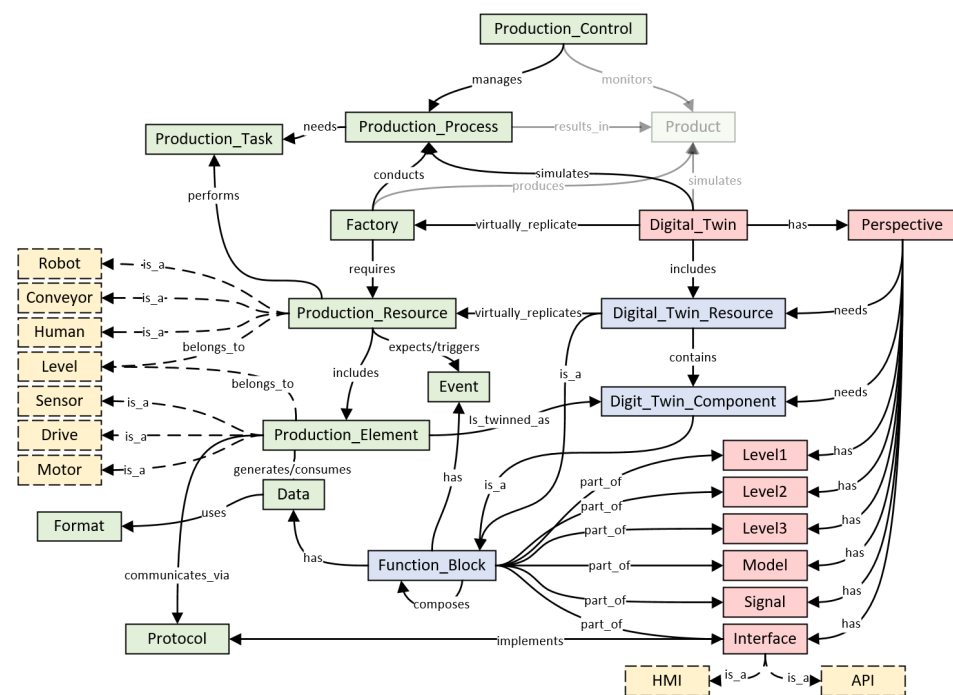


Figure 9. Digital Twin Architecture Ontology Model (DTAOM). [Source: Own figure].

5. Use Case Example

5.1. FASTory Use Case

To demonstrate the proposed approach, a production line with a discrete manufacturing use case is used. The production line is known as the FASTory assembly line where modular and sequential loop-like workstation configuration is used to mimic the production of electronic devices as depicted in Figure 10. The production system includes 10 identical workstations (labeled as WS2–WS6 and WS8–WS12) that are equipped with conveyors and different robots to perform the assembly processes. The assembly process mimics assembling mobile phones where robots draw different parts of the phone with different colors and different models to emulate the variation of the product. Workstation 1 and workstation 7 are used for pallets and material handling.

The research in [1] illustrates the physical system and the development of a simulator to help in developing applications for the production system. Furthermore, this use case was exploited in several research activities such as developing a generic MES system in [85], proposing a knowledge-based manufacturing system in [86], studying the industrial data lifecycle management in [87], and implementing the ISO-22400 KPIs in a virtual production systems in [88] among others. To reduce the complexity of demonstrating the approach in this paper, a single workstation will be used where the assembly process happens. As

shown in the Figure 10, any drawing workstation, i.e., WS2, consists of two conveyors. The main conveyor is for passing the pallet to the work cell in order to draw the phone parts. Meanwhile, the by-pass conveyor allows the pallets to move past the workstation in case the robot is in a busy state. This configuration guarantees zero blockages due to pallet congestion. Five proximity sensors are attached to the conveyors to measure the presents of the pallet in the workstation. In addition, an RFID reader is located at the beginning of each workstation to identify each arriving pallet.

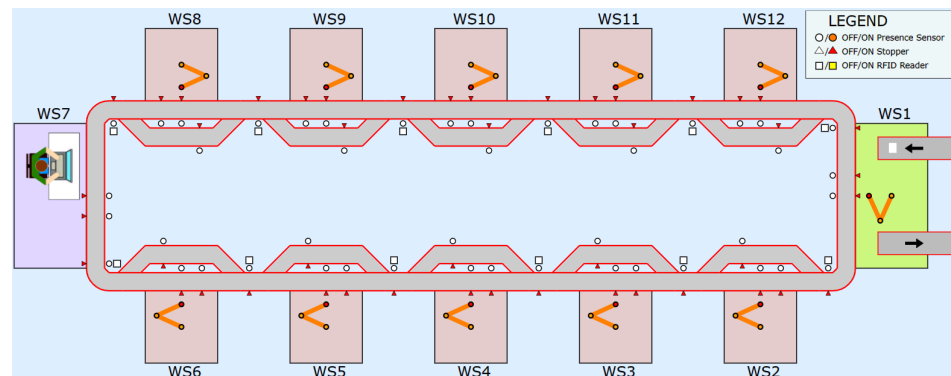


Figure 10. FASTory assembly line layout. [Source: Snapshot from the FASTory Simulator [1]].

Figure 11 depicts a graph description of the workstation process and resources. Each workstation is controlled by two Remote Terminal Units (RTUs) [89]. These RTUs provide connectivity to the MES and ERP systems using web services. Then, the Robot RTU controls the robot controller, which manipulates the robotic arm. The gripper is attached to the robotic arm, and it grasps the pen that has certain color to draw on the paper. This paper represents the product and is attached to a pallet. The pallet is transferred by the main conveyor if the robot is in an idle, stop or down state, and by the by-pass conveyor if the robot is in the busy state. These conveyors are moved by separate motors that are driven by separate drives. In addition, proximity sensors and an RFID are located at certain locations along the conveyor belts. Finally, these drives, sensors and the RFID are controlled and read by the Conveyor RTU.

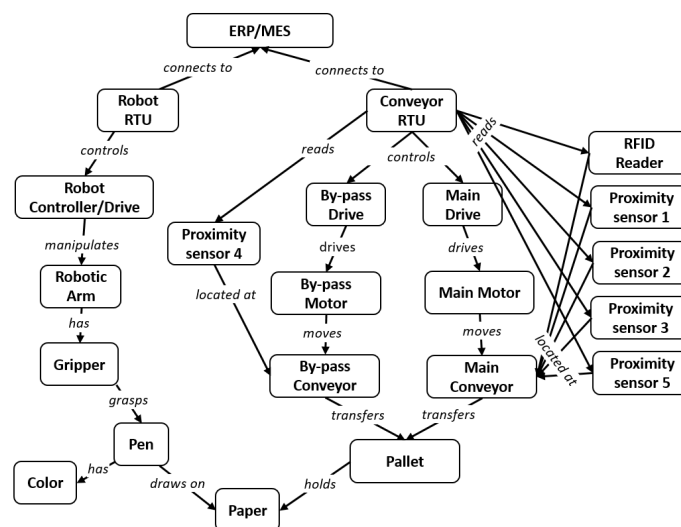


Figure 11. FASTory production resources description. [Source: Own figure].

5.2. Function Block-Based Digital Twin Architecture

For the FASTory use case, and following the graph in Figure 11, three main production resources can be listed including the robot, the main conveyor and the by-pass conveyor. Additionally, the proximity sensors, the RFID reader and the gripper can be added as

production elements. Figure 12 depicts the populated model of the process description for the FASTory use case. The figure shows that the model includes four processes where three processes are performed by the robot and one process is performed by the transportation system.

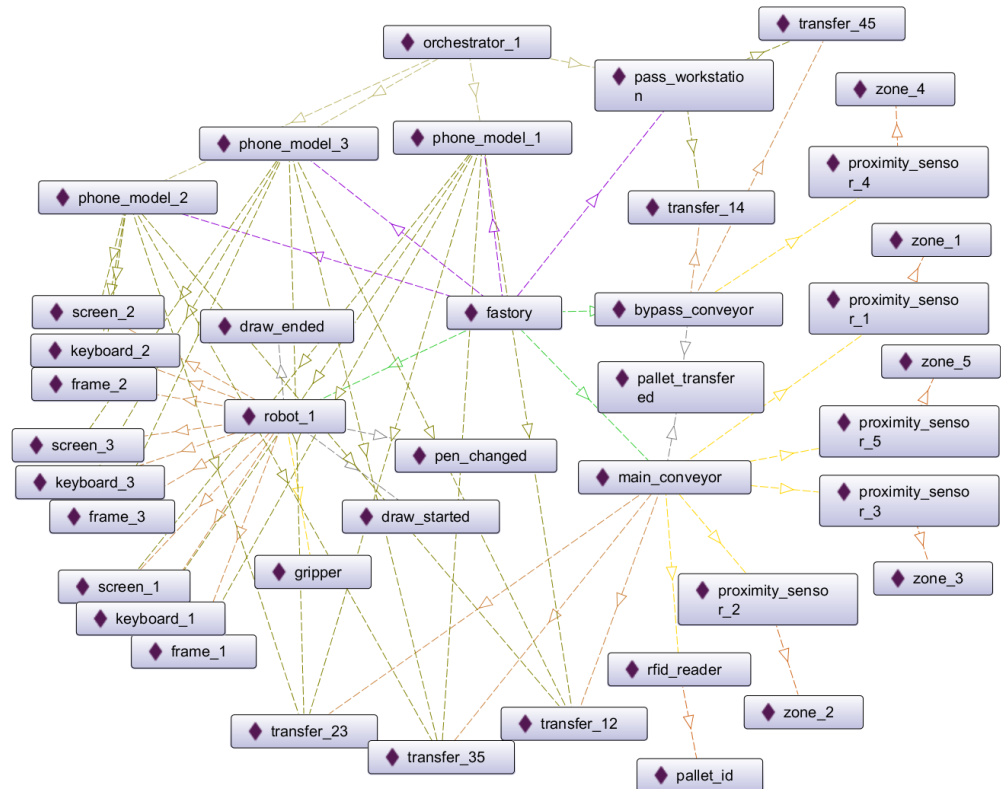


Figure 12. FASTory production description for drawing process. [Source: Own figure which is a snapshot from Protege Ontology Editor].

For the Data, six instances have been included in the use case as each zone includes the status of the proximity sensor that is related to. The *pallet_id* data instance is related to the RFID reader and it holds the pallet identifier code. For the events, four events have been added. These events include *draw_ended*, *draw_started*, *pen_changed*, which are triggered by the robot and *pallet_transferred* which is triggered by the conveyors.

Afterwards, the digital twin specifications can be introduced. For this use case, two perspectives can be created. The first one is the power consumption perspective. The second perspective can be resource utilization. Figure 13 depicts an example of a populated ontology for the digital twin specifications. As seen in the figure, each instance of the perspective class owns three instances of the model, three instances of the signals and three instances of the interface. These instances will be linked later with the digital twin components after the twinning operation. The third knowledge source is legacy knowledge. This knowledge includes all descriptions and definitions of the production resources. As an example for this use case, Figure 14 depicts a snippet of the ontology that describes the legacy knowledge of FASTory workstations as presented in Figure 11.

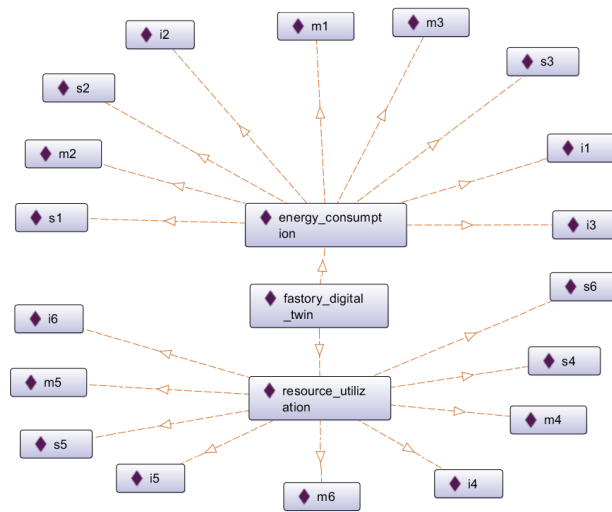


Figure 13. Twinning specifications for FASTory use case. [Source: Own figure which is a snapshot from Protege Ontology Editor].

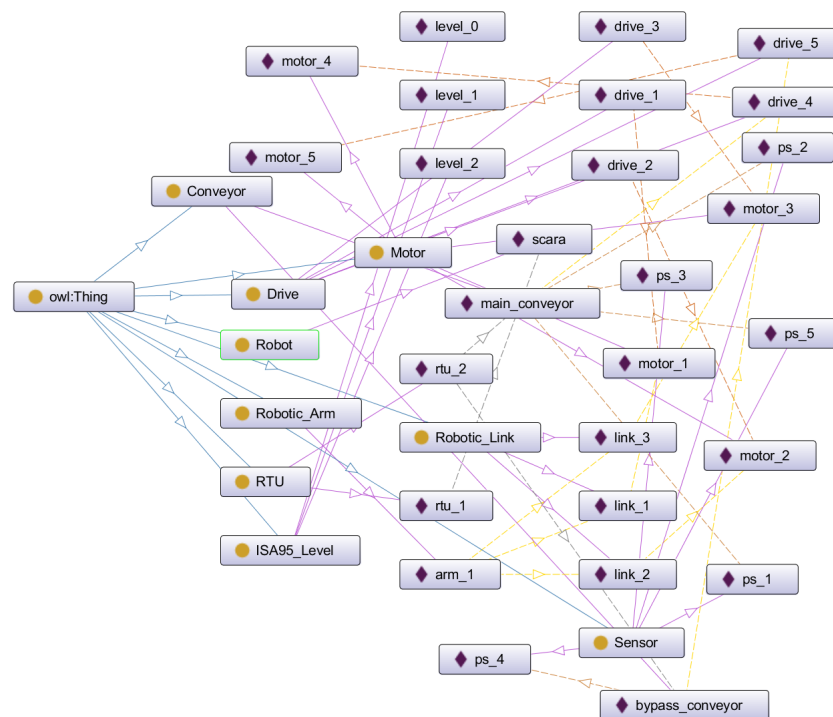


Figure 14. Snippet of the legacy ontology for the FASTory use case. [Source: Own figure which is a snapshot from Protege Ontology Editor].

Once these ontology models (seen in Figures 12–14) are populated, a set of rules are needed for the reasoning. These SWRL (Semantic Web Rule Language) rules are part of the *Digital Twin Architecture* ontology model (in blue in Figure 8). In this regard, Equation (1) is a rule that creates a *Digital_Twin_Resource* instance for each *Production_Resource*. Then, a datatype *type* is added to the *Digital_Twin_Resource* instances which highlight the virtual feature using the rule in Equation (2). Equation (3) can be used for linking the virtual and physical resources.

$$Production_Resource(?x) \implies Digital_Twin_Resource(?x) \tag{1}$$

$$Digital_Twin_Resource(?x) \implies type(?x, "virtual - resource") \quad (2)$$

$$\begin{aligned} & Production_Resource(?x) \wedge \\ & Digital_Twin_Resource(?x) \implies virtually_replicated(?x, ?x) \end{aligned} \quad (3)$$

Afterwards, an instance of *Function_Block* can be created for each instance of the *Digital_Twin_Resource* using the rule in Equation (4). Additionally, the same rule links the function block instance with the digital twin resource instance using *is* object property. Equation (5) depicts the rule for adding an alias to the function block as a data property. Then rules in Equations (6) and (7) add the events to the function blocks.

$$\begin{aligned} & Digital_Twin_Resource(?x) \wedge \\ & swrl : makeOWLThing(?y, ?x) \implies Function_Block(?y) \wedge is(?x, ?y) \end{aligned} \quad (4)$$

$$\begin{aligned} & Function_Block(?x) \wedge \\ & Production_Resource(?y) \wedge \\ & name(?y, ?z) \implies alias(?x, ?z) \end{aligned} \quad (5)$$

$$\begin{aligned} & Function_Block(?x) \wedge \\ & Production_Resource(?y) \wedge \\ & expects(?y, ?z) \implies has(?x, ?z) \end{aligned} \quad (6)$$

$$\begin{aligned} & Function_Block(?x) \wedge \\ & Production_Resource(?y) \wedge \\ & triggers(?y, ?z) \implies has(?x, ?z) \end{aligned} \quad (7)$$

Once the digital twin resources are created, the digital twin components can be created and linked. This can be seen in the rules in Equations (8)–(10).

$$Production_Element(?x) \implies Digital_Twin_Component(?x) \quad (8)$$

$$Digital_Twin_Component(?x) \implies type(?x, "virtual - element") \quad (9)$$

$$\begin{aligned} & Production_Element(?x) \wedge \\ & Digital_Twin_Component(?x) \implies is_twinned_as(?x, ?x) \end{aligned} \quad (10)$$

Afterwards, the digital twin components are represented as function blocks. Equation (11) creates the function block, while Equation (12) adds the alias to the created instance of the function blocks. Then, Equations (13) and (14) link the data input and outputs of the function block

$$\begin{aligned} & Digital_Twin_Component(?x) \wedge \\ & swrl : makeOWLThing(?y, ?x) \implies Function_Block(?y) \wedge is(?x, ?y) \end{aligned} \quad (11)$$

$$\begin{aligned} & Function_Block(?x) \wedge \\ & Production_Element(?y) \wedge \\ & name(?y, ?z) \implies alias(?x, ?z) \end{aligned} \quad (12)$$

$$\begin{aligned} & Function_Block(?x) \wedge \\ & Production_Element(?y) \wedge \\ & generates(?y, ?z) \implies has(?x, ?z) \end{aligned} \quad (13)$$

$$\begin{aligned} & Function_Block(?x) \wedge \\ & Production_Element(?y) \wedge \\ & consumes(?y, ?z) \implies has(?x, ?z) \end{aligned} \quad (14)$$

The created instances of the function block represent the digital twin components and the digital twin resources. To compose the function blocks, the rule in Equation (15) is used to combine the digital twin components and resources using the function blocks.

$$\begin{aligned}
 & Digital_Twin_Resource(?x1) \wedge \\
 & Digital_Twin_component(?x2) \wedge \\
 & \quad contains(?x1, ?x2) \wedge \\
 & \quad \quad is_a(?x1, ?x3) \wedge \\
 & \quad \quad \quad is_a(?x2, ?x4) \implies composes(?x4, ?x3)
 \end{aligned} \tag{15}$$

Lastly, the function components will be sorted to classify them using the 9-block concept. Initially, as presented in Figure 13, two perspectives are created for this digital twin. Each perspective has three layers and three levels. For the levels, the function blocks that belong to each level follow the associations of the production resources and production elements. This knowledge comes from the description of these instances which are represented in the legacy knowledge. Equation (16) depicts the rule for making this linkage.

$$\begin{aligned}
 & Digital_Twin_Resource(?x1) \wedge \\
 & \quad Production_Resource(?x1) \wedge \\
 & Digital_Twin_component(?x2) \wedge \\
 & \quad Production_Element(?x2) \wedge \\
 & \quad \quad Function_Block(?x3) \wedge \\
 & \quad \quad \quad Function_Block(?x4) \wedge \\
 & \quad \quad \quad \quad is_a(?x1, ?x3) \wedge \\
 & \quad \quad \quad \quad \quad is_a(?x2, ?x4) \wedge \\
 & \quad \quad \quad \quad \quad \quad belongs_to(?x1, ?x5) \wedge \\
 & \quad \quad \quad \quad \quad \quad \quad belongs_to(?x2, ?x6) \implies part_of(?x1, ?x5) \wedge part_of(?x2, ?x6)
 \end{aligned} \tag{16}$$

In regards to the layers, each instance included in the perspective will be linked to the three layers. This is achieved via the data property *needs*. The rules in Equations (17) and (18) present the reasoning of this connection.

$$\begin{aligned}
 & Digital_Twin_Resource(?x1) \wedge \\
 & \quad Function_Block(?x2) \wedge \\
 & \quad \quad is_a(?x1, ?x2) \wedge \\
 & Perspective(?x3) \wedge \\
 & \quad \quad needs(?x3, ?x1) \wedge \\
 & \quad \quad \quad Model(?x4) \wedge \\
 & \quad \quad \quad \quad Signal(?x5) \wedge \\
 & \quad \quad \quad \quad \quad Interface(?x6) \wedge \\
 & \quad \quad \quad \quad \quad \quad has(?x3, ?x4) \wedge \\
 & \quad \quad \quad \quad \quad \quad \quad has(?x3, ?x5) \wedge \\
 & \quad \quad \quad \quad \quad \quad \quad \quad has(?x3, ?x6) \implies part_of(?x2, ?x4) \wedge part_of(?x2, ?x5) \wedge part_of(?x2, ?x6)
 \end{aligned} \tag{17}$$

$$\begin{aligned}
& Digital_Twin_Component(?x1) \wedge \\
& \quad Function_Block(?x2) \wedge \\
& \quad \quad is_a(?x1, ?x2) \wedge \\
& \quad Prespective(?x3) \wedge \\
& \quad \quad needs(?x3, ?x1) \wedge \\
& \quad \quad \quad Model(?x4) \wedge \\
& \quad \quad \quad \quad Signal(?x5) \wedge \\
& \quad \quad \quad \quad \quad Interface(?x6) \wedge \\
& \quad \quad \quad \quad \quad \quad has(?x3, ?x4) \wedge \\
& \quad \quad \quad \quad \quad \quad \quad has(?x3, ?x5) \wedge \\
& \quad \quad \quad \quad \quad \quad \quad \quad has(?x3, ?x6) \implies part_of(?x2, ?x4) \wedge part_of(?x2, ?x5) \wedge part_of(?x2, ?x6)
\end{aligned} \tag{18}$$

After the reasoning step, the architecture of the digital twin is persisted in *Digital Twin Architecture* ontology. This architecture includes the digital twin resources and the digital twin components as function blocks. Thus, the developer then needs to create the inferences function blocks in an FB editor. In this implementation, the development of the ontology and the reasoning has been performed using the Protege ontology editor [90] and Olingvo ontology editor [91]. As Figure 15 depicts, two main production resources have been identified, the robot, and transportation system which consists of the main conveyor and the by-pass conveyor. Following the reasoning flow, each one of these production resources will be represented in the digital twin world using a digital twin resource. These two resources combine several digital twin components which are reasoned via the aforementioned rules.

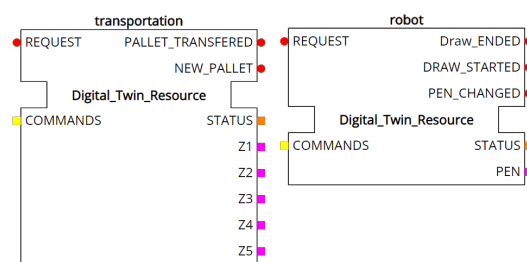


Figure 15. Two main production resources of the FASTory use case as digital twin resources. [Source: Own figure which is a snapshot from locally developed FB editor].

To simplify the presentation and satisfy the article’s length, only the transportation system is selected. Looking in more detail at the transportation resource, a total of 16 digital twin components form the transportation digital twin resource as shown in Figure 16.

The *M2* block contains the RTU which controls the conveyor’s tasks. Then, block *M1* contains the drives, motors and sensors that are related to the conveyor system. Finally, block *M0* contains the pulleys, the belts and the pallet.

At this point, the model layer has been populated, and the next stage is to look at the interface layer. As described in Section 3, the interface layer will depend on the digital twin perspectives. In this example, the digital twin has two perspectives, power consumption and resources utilization. As an example, from the power consumption perspective, the interface can include an HMI as a graph that shows the power in Watt units and APIs that use HTTP REST for publishing the consumption value. Figure 17 depicts the example for the needed function blocks for the interface and the signals.

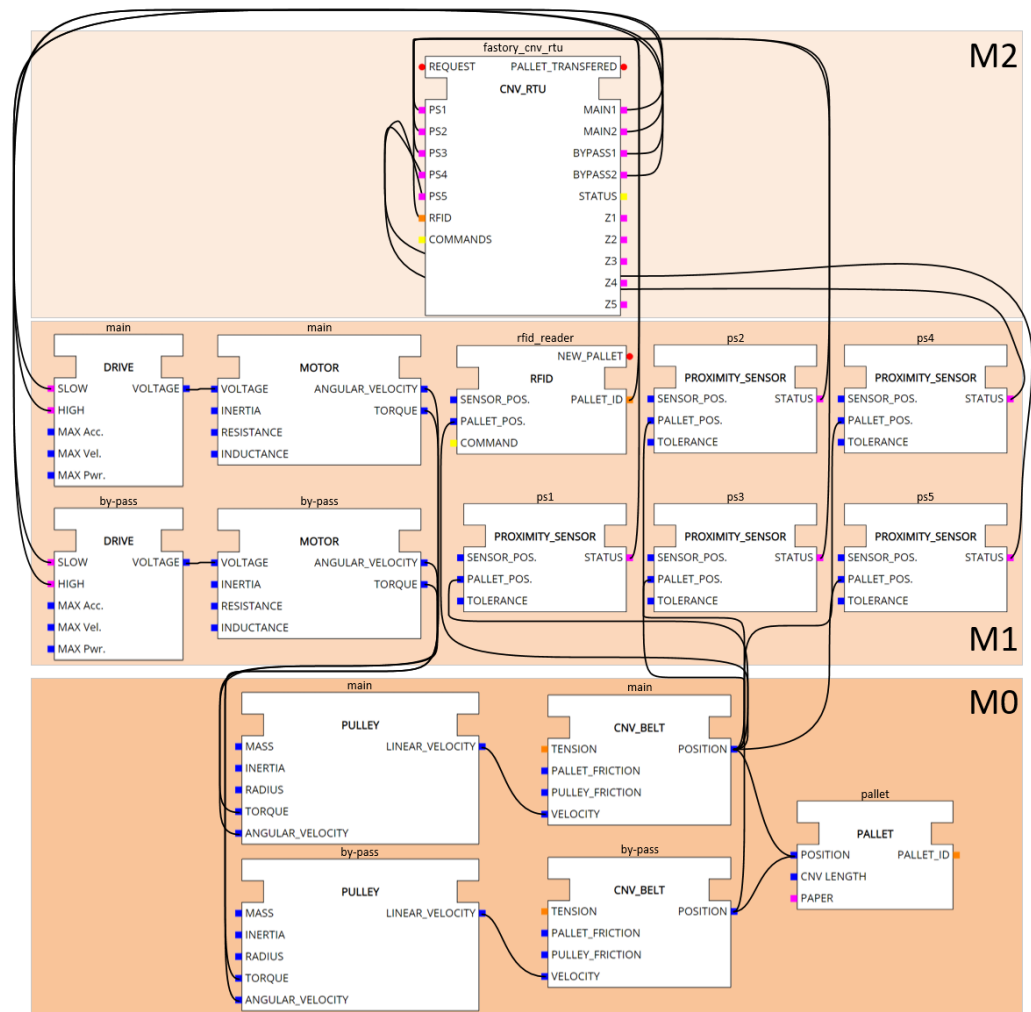


Figure 16. The composition of the transportation digital twin resource in the model layer. [Source: Own figure which is a snapshot from locally developed FB editor].

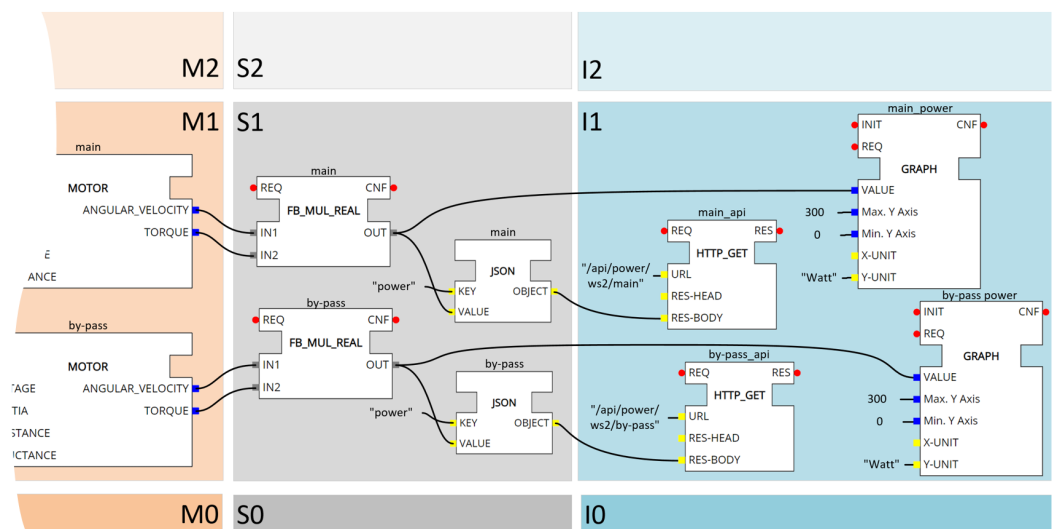


Figure 17. The signal and interface layers for the transportation digital resource in the FASTory use case. [Source: Own figure].

6. Discussion

The demonstrated approach in the previous section permits high flexibility for the developers to choose the desired components (i.e., function blocks) for the digital twin. The use case showed that the model layer is the core for building the digital twin. This layer depends mostly on the physical system descriptions including processes and equipment. Once the model layer is drafted, the interface layer can be developed to satisfy the objectives of the digital twin. This layer depends entirely on the perspectives of the digital twin and the features of the digital twin. Then, the crucially needed signal layer can be developed to connect the model layer with the interface layer. The three model levels were needed in building the digital twin for the transportation resource. However, only level 1 was needed in the signals and interfaces layers due to the fact that the shared information is at level 1. In this regard, the paradigm, which was presented in Section 3, can be implemented differently to satisfy the developers' needs. As an example, rather than having three levels for the interface, two levels can be introduced where one can be for the HMI and the other for the Machine–Machine Interface (i.e., APIs). Furthermore, the signals layer can be only one level where all conversions and calculations can be introduced. These variations are open to the developers to decide what is best for their use case.

For the qualitative attributes that are listed in Section 3, the FASTory implementation proved the expected outcomes. As an example, the *Modularity* attribute can be demonstrated by interchanging one of the function blocks, such as the controller or the drive, to a new different one. For the *Scalability*, the demonstrated use case showed the possibility of having two conveyors by replicating the associated components. The *Reusability* was evident in the use of the drive and the motor function blocks, where these two function blocks are from legacy implementation and are still possible to be reused. The *Interoperability* can be also guaranteed as the approach allows the user to use self-made function blocks or commercially available ones. In the demonstrated example, the system will work fine if the RTU is changed with a PLC or other type of controller from a different vendor. Lastly, the *Composability* is a feature of the IEC61499 standard which is included in this implementation. As an example, the transportation function block is composed of several function blocks.

The presented approach is still in its first versions and very well open for modifications and improvements. Thus, looking at the standard ISO 23247 and trying to find synergies with the presented approach is very well accepted. In this regard, both solutions dedicated separate modules for interfaces (user entity) and models (digital twin entity) which is important to guide the user in architecting such a complicated system. Both approaches addressed the information exchange as well. However, ISO 23247 did not dedicate a module for such an operation and left it open to the developers. While the presented approach considered these operations in the signal layer. The standard split the interfacing with the outside world into two entities (user and data) which might be necessary for more generic targeted domains. Overall, the presented approach appeared to be more specific for the manufacturing domain while the ISO 23247 standard targets a wider range of applications.

7. Conclusions and Future Work

Due to the versatility in exploiting them, digital twins became more essential in industrial applications. Adding to that, the benefits of saving energy, cost and time, and providing detailed insights about the behavior of systems prove to be very valuable. Thus, having a framework or paradigm will help the developers in creating such completed systems. In this regard, the conducted research provided guidelines or a paradigm for the developers to architect digital twins. The presented approach included multi-layer, multi-level and multi-perspective concepts for building a digital twin for manufacturing processes. Inspired by the ISA95 standard, the multi-level concept allows the user to allocate each component based on the level in the manufacturing pyramid. While the multi-layer concept is inspired by the onion architecture and helps the programmers and developers to draft the needed components based on the functionality. Finally, the multi-

perspective concept allows the user to distinctly build the digital twin views or use cases. This combination of the concepts permits the creation of a modular, scalable, reusable, interoperable and composable digital twin as demonstrated in the paper. Then, to help illustrate the approach, an ontology-based implementation was presented. It was clearly seen that developing a digital twin requires substantial knowledge of the physical system that needs to be twined, the technical aspects and features of the development environment and the domain where the digital twin will be exploited. After the demonstration, a comparison with the ISO 23247 standard was drafted to find synergies between the two approaches. Hence, it was observed clearly that digital twins require three main aspects; modeling, interfacing and information exchange. These three main aspects were found in both approaches but with different implementations and conceptualizations. Overall, the differences between the approach can be seen in the targeted outcomes as the standard focuses on a wider range of applications while the presented approach in this paper targets manufacturing processes specifically. In fact, this comparison opens the doors for future work which may include addressing digital twinning for products and spatial computing. Moreover, future research may include extending the ontology concepts in this research to be compatible with the IOF vocabulary.

Author Contributions: Conceptualization, W.M.M. and J.L.M.L.; methodology, W.M.M.; software, W.M.M.; validation, W.M.M.; formal analysis, W.M.M.; investigation, W.M.M.; resources, W.M.M. and J.L.M.L.; data curation, W.M.M.; writing—original draft preparation, W.M.M.; writing—review and editing, W.M.M., R.E.H. and J.L.M.L.; visualization, W.M.M.; supervision, R.E.H. and J.L.M.L.; project administration, J.L.M.L.; funding acquisition, J.L.M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The research work by Wael M. Mohammed is supported by the Doctoral School of the President call 2018 of the Tampere University of Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mohammed, W.M.; Lobov, A.; Ferrer, B.R.; Iarovyi, S.; Lastra, J.L.M. A web-based simulator for a discrete manufacturing system. In Proceedings of the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 6583–6589. <https://doi.org/10.1109/IECON.2016.7793563>.
2. Gelbart, N.R. *The King's Midwife: A History and Mystery of Madame du Coudray*; University of California Press: Berkeley, CA, USA; London, UK, 1999.
3. Hormaza, L.A.; Mohammed, W.M.; Ferrer, B.R.; Bejarano, R.; Martinez Lastra, J.L. On-line Training and Monitoring of Robot Tasks through Virtual Reality. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; pp. 841–846. <https://doi.org/10.1109/INDIN41052.2019.8971967>.
4. Jones, D.; Snider, C.; Nassehi, A.; Yon, J.; Hicks, B. Characterising the Digital Twin: A systematic literature review. *CIRP J. Manuf. Sci. Technol.* **2020**, *29*, 36–52. <https://doi.org/10.1016/j.cirpj.2020.02.002>.
5. Grieves, M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication. *White Paper* **2015**.
6. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. <https://doi.org/10.1109/TII.2018.2873186>.
7. Barricelli, B.R.; Casiraghi, E.; Fogli, D. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. *IEEE Access* **2019**, *7*, 167653–167671. <https://doi.org/10.1109/ACCESS.2019.2953499>.
8. VanDerHorn, E.; Mahadevan, S. Digital Twin: Generalization, characterization and implementation. *Decis. Support Syst.* **2021**, *145*, 113524. <https://doi.org/10.1016/j.dss.2021.113524>.
9. Guerra, R.H.; Quiza, R.; Villalonga, A.; Arenas, J.; Castano, F. Digital Twin-Based Optimization for Ultraprecision Motion Systems With Backlash and Friction. *IEEE Access* **2019**, *7*, 93462–93472. <https://doi.org/10.1109/ACCESS.2019.2928141>.
10. Grieves, M. Intelligent Digital Twins: The Role of AI and ML in the Future of Digital Twins Chief Scientist of Advanced Manufacturing Florida Institute of Technology 2021, unpublished. <https://doi.org/10.13140/RG.2.2.17464.67849>.
11. Mohammed, W.M.; Lastra, J.L.M. FASTory Digital Twin Data. *Data Brief* **2021**, *35*, 106912. <https://doi.org/10.1016/j.dib.2021.106912>.

12. Cheng, K.; Wang, Q.; Yang, D.; Dai, Q.; Wang, M. Digital-Twins-Driven Semi-Physical Simulation for Testing and Evaluation of Industrial Software in a Smart Manufacturing System. *Machines* **2022**, *10*, 388. <https://doi.org/10.3390/machines10050388>.
13. Nie, Z.; Cao, G.; Zhang, P.; Peng, Q.; Zhang, Z. Multi-Analogy Innovation Design Based on Digital Twin. *Machines* **2022**, *10*, 652. <https://doi.org/10.3390/machines10080652>.
14. Cruz, Y.J.; Rivas, M.; Quiza, R.; Haber, R.E.; Castaño, F.; Villalonga, A. A two-step machine learning approach for dynamic model selection: A case study on a micro milling process. *Comput. Ind.* **2022**, *143*, 103764. <https://doi.org/10.1016/j.compind.2022.103764>.
15. Villalonga, A.; Negri, E.; Biscardo, G.; Castano, F.; Haber, R.E.; Fumagalli, L.; Macchi, M. A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins. *Annu. Rev. Control.* **2021**, *51*, 357–373. <https://doi.org/10.1016/j.arcontrol.2021.04.008>.
16. Khan, M.G.; Huda, N.U.; Zaman, U.K.U. Smart Warehouse Management System: Architecture, Real-Time Implementation and Prototype Design. *Machines* **2022**, *10*, 150. <https://doi.org/10.3390/machines10020150>.
17. Cimino, C.; Negri, E.; Fumagalli, L. Review of digital twin applications in manufacturing. *Comput. Ind.* **2019**, *113*, 103130. <https://doi.org/10.1016/j.compind.2019.103130>.
18. Zhang, J.; Deng, T.; Jiang, H.; Chen, H.; Qin, S.; Ding, G. Bi-level dynamic scheduling architecture based on service unit digital twin agents. *J. Manuf. Syst.* **2021**, *60*, 59–79. <https://doi.org/10.1016/j.jmsy.2021.05.007>.
19. Fang, Y.; Peng, C.; Lou, P.; Zhou, Z.; Hu, J.; Yan, J. Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6425–6435. <https://doi.org/10.1109/TII.2019.2938572>.
20. Resman, M.; Pipan, M.; Simic, M.; Herakovic, N. A new architecture model for smart manufacturing: A performance analysis and comparison with the RAMI 4.0 reference model. *Adv. Prod. Eng. Manag.* **2019**, *14*, 153–165. <https://doi.org/10.14743/apem2019.2.318>.
21. Josifovska, K.; Yigitbas, E.; Engels, G. Reference Framework for Digital Twins within Cyber-Physical Systems. In Proceedings of the 2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS), Montreal, QC, Canada, 28 May 2019; pp. 25–31. <https://doi.org/10.1109/SEsCPS.2019.00012>.
22. Tekinerdogan, B.; Verdouw, C. Systems Architecture Design Pattern Catalog for Developing Digital Twins. *Sensors* **2020**, *20*, 5103. <https://doi.org/10.3390/s20185103>.
23. Li, X.; He, B.; Wang, Z.; Zhou, Y.; Li, G.; Jiang, R. Semantic-Enhanced Digital Twin System for Robot–Environment Interaction Monitoring. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. <https://doi.org/10.1109/TIM.2021.3066542>.
24. Mattila, J.; Ala-Laurinaho, R.; Autiosalo, J.; Salminen, P.; Tammi, K. Using Digital Twin Documents to Control a Smart Factory: Simulation Approach with ROS, Gazebo, and Twinbase. *Machines* **2022**, *10*, 225. <https://doi.org/10.3390/machines10040225>.
25. van der Valk, H.; Haße, H.; Möller, F.; Arbter, M.; Henning, J.L.; Otto, B. A Taxonomy of Digital Twins. *AMCIS 2020 Proc.* **2020**, *4*.
26. Redelinghuys, A.J.H.; Kruger, K.; Basson, A. A Six-Layer Architecture for Digital Twins with Aggregation. In *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*; Borangiu, T., Trentesaux, D., Leitão, P., Giret Boggino, A., Botti, V., Eds.; Series Title: Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2020; Volume 853, pp. 171–182. https://doi.org/10.1007/978-3-030-27477-1_13.
27. Resman, M.; Protner, J.; Simic, M.; Herakovic, N. A Five-Step Approach to Planning Data-Driven Digital Twins for Discrete Manufacturing Systems. *Appl. Sci.* **2021**, *11*, 3639. <https://doi.org/10.3390/app11083639>.
28. Somers, S.; Oltramari, A.; Lebiere, C. Cognitive twin: A cognitive approach to personalized assistants. In *Proceedings of the CEUR Workshop Proc*; Martin, A., Hinkelmann, K., Fill, H.-G., Gerber, A., Lenat, D., Stolle R., van Harmelen F., Eds.; CEUR-WS: Palo Alto, CA, USA, 2020; Volume 2600.
29. Zheng, X.; Lu, J.; Kiritsis, D. The emergence of cognitive digital twin: vision, challenges and opportunities. *Int. J. Prod. Res.* **2021**, *0*, 1–23. <https://doi.org/10.1080/00207543.2021.2014591>.
30. Abburu, S.; Berre, A.; Jacoby, M.; Roman, D.; Stojanovic, L.; Stojanovic, N. COGNITWIN - Hybrid and Cognitive Digital Twins for the Process Industry. In Proceedings of the 2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Cardiff, UK, 15–17 June 2020. <https://doi.org/10.1109/ICE/ITMC49519.2020.9198403>.
31. Lu, J.; Zheng, X.; Gharaei, A.; Kalaboukas, K.; Kiritsis, D. Cognitive Twins for Supporting Decision-Makings of Internet of Things Systems. In *Proceedings of 5th International Conference on the Industry 4.0 Model for Advanced Manufacturing*; Wang L., Majstorovic, V.D., Mourtzis D., Carpanzano E., Moroni G., Galantucci L.M.; Springer International Publishing: Cham, Switzerland, 2020; pp. 105–115. https://doi.org/10.1007/978-3-030-46212-3_7.
32. Rožanec, J.; Lu, J.; Rupnik, J.; Škrjanc, M.; Mladenčić, D.; Fortuna, B.; Zheng, X.; Kiritsis, D. Actionable cognitive twins for decision making in manufacturing. *Int. J. Prod. Res.* **2022**, *60*, 452–478. <https://doi.org/10.1080/00207543.2021.2002967>.
33. Rožanec, J.; Jinzhi, L.; Košmerlj, A.; Kenda, K.; Dimitris, K.; Jovanoski, V.; Rupnik, J.; Karlovčec, M.; Fortuna, B. Towards actionable cognitive digital twins for manufacturing. In *Proceedings of the CEUR Workshop Proc*; Garcia-Castro, R., Davies, J., Antoniou, G., Fortuna, C., Eds.; CEUR-WS: Palo Alto, CA, USA, 2020; Volume 2615.
34. Li, H.; Wang, G.; Lu, J.; Kiritsis, D. Cognitive twin construction for system of systems operation based on semantic integration and high-level architecture. *Integr. -Comput. -Aided Eng.* **2022**, *29*, 277–295. <https://doi.org/10.3233/ICA-220677>.
35. digitaltwin.io. Digital Twin Knowledge Repository-Platforms. Available online: <https://digitaltwin.io/platforms.html> (accessed on 16 February 2021).
36. General Electric. Digital Twin | GE Digital. 2021. Available online: <https://www.ge.com/digital/applications/digital-twin> (accessed on 16 February 2021).

37. Colin Parris. What is a Digital Twin? Available online: <https://www.ge.com/digital/blog/what-digital-twin> (accessed on 29 August 2022).
38. IBM. Exchange | IBM Digital Twin Exchange. Available online: <https://digitaltwinexchange.ibm.com/exchange/> (accessed on 16 February 2021).
39. PTC. What Is Digital Twin? | PTC. Available online: <https://www.ptc.com/en/industry-insights/digital-twin> (accessed on 16 February 2021).
40. McMahon, C.; Dertien, S. State of Digital Twin. Available online: <https://www.ptc.com/-/media/Files/PDFs/Manufacturing/State-of-Digital-Twin-2022.pdf> (accessed on 22 August 2022).
41. Microsoft. Azure Digital Twins. Available online: <https://sourceforge.net/software/product/Azure-Digital-Twins/> (accessed on 16 February 2021).
42. baanders. Data history (with Azure Data Explorer)-Azure Digital Twins. Available online: <https://docs.microsoft.com/en-us/azure/digital-twins/concepts-data-history> (accessed on 22 August 2022).
43. Microsoft. Digital Twins—Modeling and Simulations | Microsoft Azure. Available online: <https://azure.microsoft.com/en-us/services/digital-twins/> (accessed on 18 February 2021).
44. Ansys. Twin Builder: Digital Twin Predictive Maintenance Software | Ansys. Available online: <https://www.ansys.com/products/systems/ansys-twin-builder> (accessed on 16 February 2021).
45. Ansys Inc. White Paper on Digital Twins: Making the Vision Achievable. Available online: <https://www.ansys.com/resource-center/white-paper/digital-twins-making-vision-achievable> (accessed on 22 August 2022).
46. Ansys Inc. White Paper on How Simulation-Based Digital Twins and the Industrial Internet of Things Can Improve Product and Process Performance. Available online: <https://www.ansys.com/resource-center/white-paper/how-sim-based-digital-twins-and-the-industrial-iiot-can-improve-product-and-process-perf> (accessed on 18 February 2021).
47. SAP. SAP Digital Twin Software & Technology. Available online: <https://www.sap.com/finland/products/supply-chain-management/digital-twin.html> (accessed on 16 February 2021).
48. SAP Leonardo. The Network of Digital Twins. Available online: <https://community.sap.com/topics/digital-supplier-network> (accessed on 22 August 2022).
49. Oracle Corporation. About the IoT Digital Twin Framework. Available online: <https://docs.oracle.com/en/cloud/paas/iot-cloud/iotgs/iot-digital-twin-framework.html> (accessed on 16 February 2021).
50. Bosch. When digitalization shows us the way. Available online: <https://www.boschbuildingsolutions.com/xc/en/news-and-stories/digitalization/> (accessed on 16 February 2021).
51. Emerson. Digital Twin Solutions | Emerson US. Available online: <https://www.emerson.com/en-us/automation/operations-business-management/dynamic-simulation/digital-twin-solutions> (accessed on 16 February 2021).
52. Emerson. Emerson Digital Twin: A Key Technology for Digital Transformation. Available online: <https://www.emerson.com/documents/automation/white-paper-emerson-digital-twin-a-key-technology-for-digital-transformation-mimic-en-5262472.pdf> (accessed on 22 August 2022).
53. ABB. Digital twin applications-What is new | ABB. Available online: <https://new.abb.com/control-systems/features/digital-twin-applications> (accessed on 16 February 2021).
54. Mathworks. Digital Twins for Predictive Maintenance-04. Available online: <https://explore.mathworks.com/digital-twins-for-predictive-maintenance/chapter-04-36US-761YW.html> (accessed on 22 August 2022).
55. Mathworks. Digital Twins for Predictive Maintenance. Available online: <https://explore.mathworks.com/digital-twins-for-predictive-maintenance/chapter-05-36US-763YW.html> (accessed on 16 February 2021).
56. COMSOL Multiphysics® Software-Understand, Predict, and Optimize. Available online: <https://www.comsol.com/comsol-multiphysics> (accessed on 24 February 2021).
57. NVIDIA. NVIDIA Omniverse for Digital Twins. Available online: <https://www.nvidia.com/en-us/omniverse/solutions/digital-twins/> (accessed on 18 August 2022).
58. NVIDIA. Develop on NVIDIA Omniverse Platform. Available online: <https://developer.nvidia.com/nvidia-omniverse-platform> (accessed on 22 August 2022).
59. Visual Components. Digital Twins and Virtual Commissioning in the Manufacturing Industry (Updated for 2022). Available online: <https://www.visualcomponents.com/resources/blog/digital-twins-and-virtual-commissioning-in-industry-4-0/> (accessed on 6 September 2022).
60. Siemens. Tecnomatix | Siemens Software. Available online: <https://www.plm.automation.siemens.com/global/en/products/tecnomatix/> (accessed on 6 September 2022).
61. iconics. Make Better Decisions About Smart Building Operations with Digital Twin Technology | ICONICS Software Solutions. Available online: <https://iconics.com/Resources/ICONICS-Blog/2022/Make-Better-Decisions-About-Smart-Building-Operations-with-Digital-Twin-Technology> (accessed on 22 August 2022).
62. Microsoft. Pushing the boundaries of renewable energy production efficiency with Azure Digital Twins. Available online: <https://customers.microsoft.com/en-in/story/848311-dooan-manufacturing-azure-digital-twins> (accessed on 22 August 2022).
63. Visual Components. Product Overview-Visual Components 4.5. Available online: <https://www.visualcomponents.com/products/> (accessed on 6 September 2022).

64. Foundation, B. Blender is for Simulation. Available online: <https://www.blender.org/features/simulation/> (accessed on 22 August 2022).
65. Maya Software | Get Prices and Buy Maya 2023 | Autodesk. Available online: <https://www.autodesk.com/products/maya/overview> (accessed on 22 August 2022).
66. Caulfield, B. NVIDIA, BMW Blend Reality, Virtual Worlds to Demonstrate Factory of the Future. Available online: <https://blogs.nvidia.com/blog/2021/04/13/nvidia-bmw-factory-future/> (accessed 22 August 2022).
67. Wang, K.; Wang, Y.; Li, Y.; Fan, X.; Xiao, S.; Hu, L. A review of the technology standards for enabling digital twin. *Digit. Twin* **2022**, *2*, 4. <https://doi.org/10.12688/digitaltwin.17549.1>.
68. International Organization for Standardization (ISO). ISO 23247 Automation systems and integration—Digital twin framework for manufacturing. Available online: <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/50/75066.html> (accessed on 22 August 2022).
69. Shao, G. Use Case Scenarios for Digital Twin Implementation Based on ISO 23247. Available online: <https://doi.org/10.6028/NIST.AMS.400-2> (accessed on 22 August 2022).
70. International Organization for Standardization (ISO). ISO/TR 24464:2020 Automation systems and integration—Industrial data—Visualization elements of digital twins. <https://www.iso.org/standard/78836.html> (accessed on 22 August 2022).
71. International Organization for Standardization (ISO). ISO/IEC AWI 30172 Digital Twin—Use cases. Available online: <https://www.iso.org/standard/81578.html> (accessed on 22 August 2022).
72. International Organization for Standardization (ISO). ISO/IEC AWI 30173 Digital twin — Concepts and terminology. Available online: <https://www.iso.org/standard/81442.html> (accessed on 22 August 2022).
73. Andika, L.; Dyck, A.; Lichter, H. Towards A Design for An Extendable Reporting Interface. *Procedia Comput. Sci.* **2017**, *116*, 318–325. <https://doi.org/10.1016/j.procs.2017.10.082>.
74. Khalil, M.E.; Ghani, K.; Khalil, W. Onion architecture: A new approach for XaaS (every-thing-as-a service) based virtual collaborations. In Proceedings of the 2016 13th Learning and Technology Conference (L&T), Jeddah, Saudi Arabia, 10–11 April 2016; pp. 1–7. doi:10.1109/LT.2016.7562859.
75. Jeffrey Palermo. The Onion Architecture: Part 1. Available online: <https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/> (accessed on 22 November 2021).
76. Robert C. Martin. The Clean Architecture. Available online: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> (accessed on 22 November 2021).
77. Martin, R.C. *Clean Architecture: A Craftsman’S Guide to Software Structure and Design*, 1st Ed.; Martin series; Prentice Hall: London, UK, 2018.
78. ISA (Society). *Enterprise-Control System Integration. Models and Terminology*; ISA: Research Triangle Park, NC, USA, 2000.
79. IOF. Technical Principles–IOF. Available online: <https://www.industrialontologies.org/technical-principles/> (accessed on 6 September 2022).
80. J Neil Otte. IOF-BFO. Available online: <https://github.com/NCOR-US/IOF-BFO> (accessed on 6 Septemebr 2022).
81. Kulvatunyou, B.S.; Wallace, E.; Kiritsis, D.; Smith, B.; Will, C. The Industrial Ontologies Foundry Proof-of-Concept Project. In Proceedings of the APMS 2018 International Conference Advances in Production Management Systems (APMS 2018), Seoul, Korea, 26–30 August 2018.
82. Garetti, M.; Fumagalli, L.; Negri, E. Role of Ontologies for CPS Implementation in Manufacturing. *Manag. Prod. Eng. Rev.* **2015**, *6*, 26–32. <https://doi.org/10.1515/mper-2015-0033>.
83. Ontologies-EU Vocabularies-Publications Office of the EU. Available online: <https://op.europa.eu/en/web/eu-vocabularies/ontologies> (accessed on 14 June 2022).
84. Hyvönen, E.; Tuominen, J.; Alonen, M.; Mäkelä, E. Linked Data Finland: A 7-star Model and Platform for Publishing and Re-using Linked Datasets. In *The Semantic Web: ESWC 2014 Satellite Events*; Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8798, pp. 226–230. https://doi.org/10.1007/978-3-319-11955-7_24.
85. Mohammed, W.M.; Ramis Ferrer, B.; Iarovyi, S.; Negri, E.; Fumagalli, L.; Lobov, A.; Martinez Lastra, J.L. Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 262–274. <https://doi.org/10.1080/0951192X.2017.1407874>.
86. Iarovyi, S.; Mohammed, W.M.; Lobov, A.; Ferrer, B.R.; Lastra, J.L.M. Cyber-Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems. *Proc. IEEE* **2016**, *104*, 1142–1154. <https://doi.org/10.1109/JPROC.2015.2509498>.
87. Ramis Ferrer, B.; Mohammed, W.M.; Ahmad, M.; Iarovyi, S.; Zhang, J.; Harrison, R.; Martinez Lastra, J.L. Comparing ontologies and databases: a critical review of lifecycle engineering models in manufacturing. *Knowl. Inf. Syst.* **2021**, *63*, 1271–1304. <https://doi.org/10.1007/s10115-021-01558-4>.
88. Ramis Ferrer, B.; Muhammad, U.; Mohammed, W.; Martínez Lastra, J. Implementing and Visualizing ISO 22400 Key Performance Indicators for Monitoring Discrete Manufacturing Systems. *Machines* **2018**, *6*, 39. <https://doi.org/10.3390/machines6030039>.
89. Ferrer, B.R.; Mohammed, W.M.; Chen, E.; Lastra, J.L.M. Connecting web-based IoT devices to a cloud-based manufacturing platform. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 8628–8633. <https://doi.org/10.1109/IECON.2017.8217516>.

90. Stanford. Protégé-A free, open-source ontology editor and framework for building intelligent systems. Available online: <https://protege.stanford.edu/products.php#desktop-protege> (accessed on 6 September 2022).
91. FAST-Lab, Tampere University. Olingvo—An application for editing RDF-Based ontologies. Available online: <https://www.zdmp.eu/iprdocuments/subcall/Olingvo> (accessed on 6 September 2022).