Digitization of Industrial Environments through an Industry 4.0 Compliant Approach

Lucas Sakurada*, Fernando De la Prieta†, and Paulo Leitao*‡
* Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança,
Campus de Santa Apolónia, 5300-253 Bragança, Portugal
Email: {lsakurada, pleitao}@ipb.pt
‡ Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
† BISITE Digital Innovation Hub, University of Salamanca, Edificio Multiusos I+D+i, 37007, Salamanca, Spain
Email: fer@usal.es

IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society | 979-8-3503-3182-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/IECON51785.2023.10312428

***Abstract*—About a decade after the introduction of Industry 4.0 (I4.0) as a paradigm oriented towards the digitization of industrial environments, centered on the concept of industrial Cyber-physical Systems (CPS) to enable the development of intelligent and distributed industrial systems, many companies around the world are still not immersed in this digital transformation era. This transition is not straightforward and requires the aligned with the novel technologies, architectures and standards to migrate entire traditional systems into I4.0 systems. In this context, this paper presents an approach to perform the digitization of non-I4.0 components/systems into I4.0 through an approach based on the Asset Administration Shell (AAS), which is a standardized digital representation of an asset. This approach enables to hold the asset information throughout its lifecycle, provides a standard communication interface with the asset, and is based on a set of modules that are combined with the AAS to provide novel functionalities for the asset, e.g., monitoring, diagnosis and optimization. Moreover, this approach adopts Multi-agent Systems (MAS) to provide mainly autonomy and collaborative capabilities to the system. The agents are able to get information from the AASs, making intelligent decisions and perform distributed tasks following interaction strategies, e.g., collaboration, negotiation and self-organization. The feasibility of the proposed approach was tested by digitizing a small-scale production system comprising several assets.**

Keywords: Asset Administration Shell, Industrial Cyber-Physical Systems, Industry 4.0, Multi-agent systems.

I. Introduction

In the last few years, the traditional industrial systems are becoming increasingly inadequate to meet the fast-changing market conditions, the increased global competition and the rapid technological development [1], [2]. Under this perspective, Industry 4.0 (I4.0) is promoting the digitization of industrial environments towards industrial Cyber-physical Systems (CPS) to fulfill these requirements. In this context, a set of specifications for digitizing industrial systems is offered by the Reference Architecture Model Industrie 4.0 (RAMI4.0) [3], a 3-D reference architecture that provides guidelines for engineering I4.0 systems.

Currently, the concept of Asset Administration Shell (AAS) [4] is gaining more ground in the I4.0 context, particularly enabling the effective implementation of RAMI4.0-compliant solutions. The AAS is a standardized digital representation of an asset (i.e., any logical or physical object that has value for the industry) throughout its lifecycle, which is perceived as an enabler to develop Digital Twins in a standardized manner, promoting the interoperability across different suppliers' solutions [5]. As illustrated in Figure 1, the AAS is comprised of several submodels, where the asset information, e.g., characteristics, properties and capabilities, is stored in a standard manner, being classified as passive (type 1), reactive (type 2) and proactive (type 3).

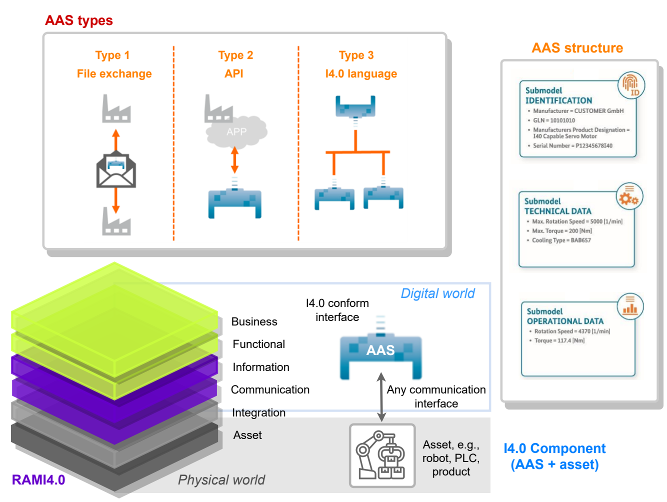

Figure 1. AAS connecting an asset into the digital world; AAS types. (1) Passive AAS. (2) Reactive AAS. (3) Proactive AAS; AAS structure.

The passive AAS, specified in [5], acts as a static file that holds the asset information along its lifecycle and can be exchanged digitally across the I4.0 network. On the other hand, the reactive AAS, specified in [6], acts as an API that responds to external requests, but does not have the capability to take initiatives and decisions. Such API enables the online access to the asset information and can be specified in a technology-neutral way, such as HTTP/REST, MQTT and OPC UA. For instance, the authors in [7], [8], implement an AAS solution focused on the plug-and-produce strategy, describing the asset information using the AutomationML in [7] and using the AASX Package Explorer tool in [8], and promoting the interaction with the assets via OPC UA communication protocol.

Finally, the proactive AAS is defined as a decision-making entity that interacts with others AASs autonomously to exchange information following the I4.0 language designed in VDI/VDE 2193. Despite there are some research works that present an interaction based on the I4.0 language [9], some aspects, namely intelligence, autonomy and collaborative capabilities in the AAS context, are usually not addressed. In this perspective, some research works are investigating the Multi-agent systems (MAS) as a key enabler technology to implement the proactive AAS itself or to support/extend its functionalities, particularly to provide intelligence, collaborative capabilities, and decision making mechanisms [10]–[12].

In recent years, several European R&D projects, namely ARUM [13], IDEAS [14], PRIME [15] and GO0D MAN [16], have demonstrated the benefits of introducing MAS in industrial environments, particularly offering alternatives to overcome the typical problems presented by the traditional centralized control approaches, which are not able to address the flexibility, robustness and reconfigurability imposed on the current industrial systems. More recently, the introduction of the industrial agent paradigm [17] supported by the recent IEEE 2660.1-2020 standard [18], have been leveraging the adoption of MAS to attend the emerging industrial challenges and creating new research opportunities in this field.

Having this is mind, this paper focuses on presenting an approach to perform the digitization of production systems not fully positioned with the I4.0 technologies, architectures and standards into I4.0 compliant systems, particularly combining the AAS and MAS. The feasibility of the proposed approach was demonstrated by digitizing a small-scale production system comprising several assets.

The rest of the paper is organized as follows. Section II describes the proposed approach to digitize non-I4.0 components/systems into I4.0, particularly using AAS and MAS. Section III presents the deployment and operation of the proposed approach into a small-scale production system. Section IV discusses the obtained outcomes. Finally, Section V rounds up the paper with the conclusions and future work.

II. I4.0 Compliant Digitization Approach

As illustrated in Figure 2, the proposed approach aims to provide guidelines and strategies for the digitization process of production systems not fully positioned with the I4.0 technologies, architectures and standards, in order to be able to make this transition seamlessly by implementing different levels of digitization. The idea is not to replace entire non-I4.0 systems, but to integrate the proposed approach as a top layer into the production system that is already in operation. Also, it is not mandatory to implement all defined levels, being their implementation dependent on available resources and requirements.

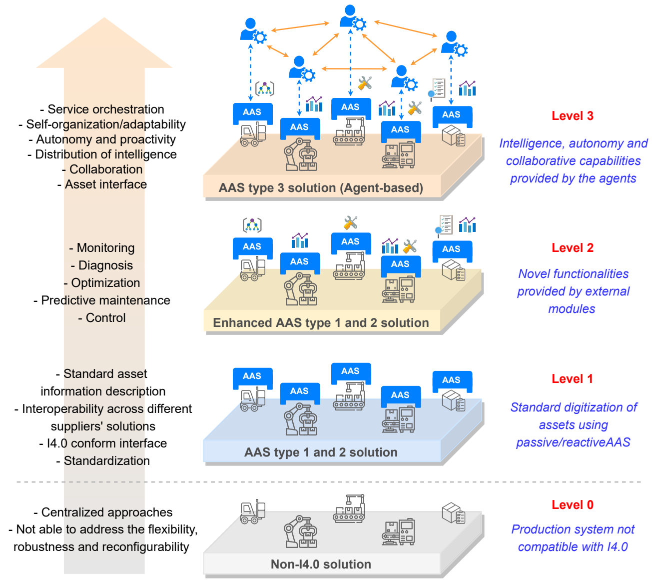

Figure 2. Overview of the I4.0 compliant digitization approach.

A. *Level 0 - Non-I4.0 solution*

The *Level 0* corresponds to the entry level, representing the traditional production system in operation, which is not fully compatible with I4.0 technologies, architectures and standards. In general, these systems present a rigid and centralized architecture that are not able to respond quickly to unexpected events and adapt to the demands of an ever-changing market.

In contrast to the dynamic nature of I4.0, *Level 0* systems have limited connectivity, intelligence and autonomy. The data exchange and integration between different components within these systems are often complex and time-consuming, leading to delays in capturing and processing data, errors, and inefficiencies in accessing and utilizing data in real-time for effective decision making. To overcome these limitations and embrace I4.0 principles, companies need to transition from *Level 0* to higher levels, particularly providing a standard digital representation of their assets.

B. *Level 1 - AAS type 1 and 2 solution*

The *Level 1* aims to digitize the assets using AASs. All the relevant asset information must be described and structured in proper submodels, as well as updated with new information along the different lifecycle phases of the digitalized asset, from design and manufacturing to operation and maintenance. At this level, AASs type 1 and 2 are considered, since the main goal is to structure the asset information in a standard manner, aiming to enable the seamless integration and interoperable communication with other systems from operational to the business level.

This level provides valuable information to be used by the upper levels of this approach. The digitization of the asset information following the AAS metamodel contributes to provide a comprehensive and transparent view of the assets, leading to increased operational efficiency and impact on various business processes. For instance, by analyzing the asset information contained in the AASs, e.g., asset performance, maintenance costs, and resource utilization, companies can identify opportunities to reduce costs, optimize asset usage, and improve overall profitability.

Regarding the development of AASs, there are several open source platforms based on the AAS specifications [5], [6] that can be used to develop the AASs type 1 and 2, namely FA^{3}ST, AASX Package Explorer tool & AASX Server, Eclipse BaSyx and Eclipse AAS Model for Java [19].

C. Level 2 - Enhanced AAS type 1 and 2 solution

The *Level 2* focuses on a modular approach to provide novel functionalities to the asset, such as monitoring, diagnosis, simulation, and other novel functionalities based on data analysis or AI techniques. Aiming to ensure compatibility and interoperability, external modules are referenced and described within the AAS common submodel following the AAS specifications. The modules are implemented as decoupled software applications, which can be designed and developed independently from the AAS development platform, such as the AASX Package Explorer tool, and integrated into the existing AAS ecosystem (see Figure 3).

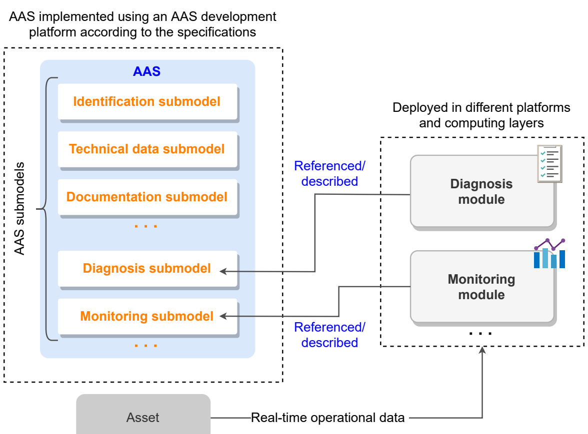

Figure 3. Decoupled modules integrated to the AAS.

As aforementioned, this level intends to provide novel functionalities (e.g., monitoring, diagnosis and simulation) for the assets, which generally depends on real-time operational data or previous data history generated by the assets. In this context, assets that does not provide operational data due to software and hardware constraints can be equipped with sensors and single-board computers. By adding these components, alternatives are offered to collect operational data and monitor various parameters and conditions of the assets. Additionally, this solution is suitable for small and medium-sized companies, as they often have limited resources compared to larger organizations, offering a cost-effective solution that can be implemented without significant changes to the system infrastructure.

D. Level 3 - AAS type 3 solution (Agent-based)

The *Level 3* offers a decentralized configuration to the original system architecture that can be applied in specific scenarios, particularly in the context of intelligent and autonomous production systems, where different assets represented by agents interact autonomously with each other to complete some production process.

In this approach, agents follow the Service-Oriented Architecture (SOA) principles, assuming the role of service provider and/or consumer. In such a setting, the main interest is the value of the offered service, being fundamental to consider service registration and discovery mechanisms to support the seamless reconfiguration. In this sense, each agent is associated with an AAS and an asset, and registers the capabilities (e.g., transportation, pick-and-place and drilling) of its asset as services in a catalogue of services, where all other agents can search for those services and request them. The information regarding of the asset's capabilities is described in the AAS submodels. Therefore, the agents need to be able to interface with the AASs to get this information. Additionally, each submodel holds specific information of the asset, which will be used by the agents as knowledge representation. The agents' knowledge representation comprises all the information that the agent has about the environment in which it is operating and/or the object that is representing. Based on this information, agents know their goals and how they should make decisions.

Bearing this in mind, the crucial point when adopting a MAS approach is the specification of the agents architecture. The use of Agent-Oriented Software Engineering (AOSE) methodologies are a suitable software engineering approach to guide the development of systems consisting of autonomous agents. Several promising AOSE methodologies have been proposed in the last years, namely Gaia, TROPOS, Prometheus, GORMAS and INGENIAS. Despite the several benefits provided by the AOSE methodologies [20], [21], their adoption in agent-based CPS solutions are not common. This fact can be verified with the help of the survey performed by the authors in [22], where MAS solutions for smart production were listed. When analyzing all these solutions, it is observed that there is no adherence in the adoption of the AOSE methodology by them. However, common patterns to design and develop these solutions were identified. In general, they concern with aspects related to identification of agents, definition of their behaviors, ontologies and interaction protocols, which are an alternative approach focusing on CPS solutions based on MAS to the traditional methodologies.

III. Digitization Example

The proposed approach was applied by digitizing a small-scale production system (see Figure 4) comprising several assets, namely two punching machines responsible for performing punching functions, two indexed line machines responsible for performing milling and drilling functions, and one transport robot IRB 1400 ABB responsible for performing the transportation of the products between the machines.

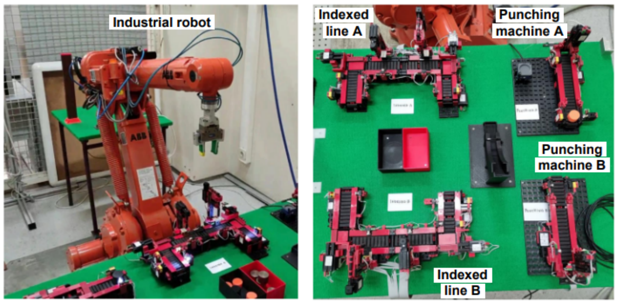

Figure 4. Small-scale production system.

A. Level 0 - Non-I4.0 solution

Initially, this small-scale production system represents the *Level 0* of the digitization process, since the system was originally designed based on a monolithic and rigid architecture controlled by an industrial controller, which is not able to address the flexibility, robustness and reconfigurability. The system is only capable of accessing the controller's memory registers to write/read variable values, in order to start/stop the operation of a machine or obtain the status of a sensor/actuator.

This system is not compliant with industry standards and does not provide any I4.0 conform interface to obtain static or operational data from the assets throughout their lifecycle (*Level 1*), it is not capable of performing intelligent functions based on data analysis or AI techniques (*Level 2*), and it lacks autonomy and proactivity to perform collaborative tasks or self-organization when necessary (*Level 3*).

B. Level 1 - AAS type 1 and 2 solution

Aiming to develop the AASs for the assets of the system, the AASX Package Explorer tool (https://github.com/admin-shell-io/aasx-package-explorer) was adopted. This tool follows the AAS specifications and enables to create/edit the AAS type 1 (file-based). Here, all the relevant information of the assets are included in the AAS submodels. For instance, Figure 5 illustrates one screenshot of the AAS created for one of the indexed line machines.

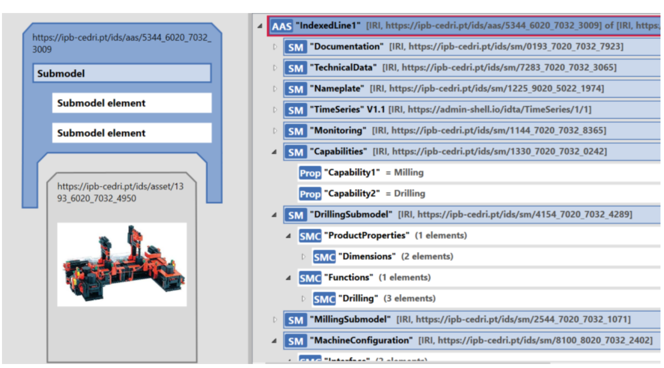

Figure 5. Screenshot of the AAS Type 1 created using the AASX Package Explorer for the indexed line.

As the AASX Package Explorer tool enables to export the AAS as a file (e.g., .aasx, .json or .xml), the AASX Server (https://github.com/admin-shell-io/aasx-server) was used in a complementary way. The AASX Server is aligned with the AAS type 2, since it automatically parse the file-based AASs and provide the asset information using several communication protocols, namely HTTP/REST, OPC UA and MQTT. For instance, by executing an HTTP/REST GET method `/aas/{aas-id}/submodels/submodelName`, it is possible to obtain the information of a specific submodel.

C. Level 2 - Enhanced AAS type 1 and 2 solution

Taking into account that the selected assets are absent of any functionality that supports the storage of the historical operational data and monitoring their health condition during the operation, a historical data and monitoring module were implemented. The historical data module interfaces with the asset via modbus and stores the data related to the production process, e.g., the processing time of each product (i.e., the period it takes from when a product starts a process on a machine and ends) in a database. In this context, the official standardized submodel called "TimeSeries" [23] was adopted to reference the database and describe how to query the asset data.

On the other hand, the monitoring module is based on a set of rules to detect trends and abnormal situations during the process, which considers the means value ($\bar{X}$) and the standard deviation (σ) to determine if a measured variable is under control or not. In this regard, the data is obtained from the database (referenced in the "TimeSeries" submodel) and analyzed based on a set of rules (see some examples in Figure 6).

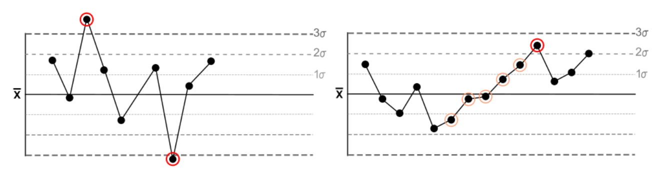

Figure 6. Implemented monitoring module. (Left) Rule 1: detection of an outlier in the evolution of the measured variable over time. (Right) Rule 2: identification of a trend in the measured variable over time.

D. Level 3 - AAS type 3 solution (Agent-based)

The agents were implemented using the FIPA-compliant Java Agent DEvelopment (JADE) framework (https://jade.tilab.com/). Regarding its features, JADE can be easily deployed in different computational platforms, providing advanced mechanisms to support the development of MAS for general applications, namely agent intercommunication mechanisms, yellow-pages service, a library of FIPA interaction protocols and a set of useful tools to debug the developed agents, e.g., Dummy, Sniffer and Introspector. In this context, two types of agents were created, namely product agent (PA) representing the products and

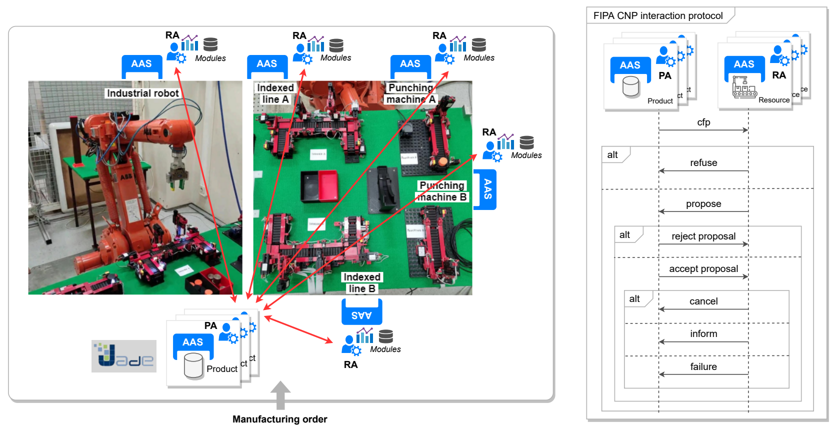

Figure 7. (Left) Deployment of the proposed approach into a small-scale production system. (Right) Interaction between the agents.

resource agent (RA) representing the two punching machines, two indexed lines and the industrial robot.

Figure 7 overviews the implementation of the proposed approach, where each asset in the system has its own AAS and agent (PA or RA). The PA gets the information from the AAS submodels (using HTTP/REST) to know the process plan of its associated product, and the RA to know the capability provided by its asset. Based on that, the agents interact autonomously following the FIPA CNP interaction protocol, aiming to complete a manufacturing order.

IV. Outlooks and Discussions

The deployment and operation of the proposed solution into a small-scale production system (*Level 0*) allowed the extraction of several outcomes. An experimental test was performed by introducing manufacturing orders in the system. These orders required the same type of product, but in different quantities, where each product is obtained by introducing a raw material in a punching machine and moving the resulting unfinished product to an indexed line, and the robot carries out the movements between the machines. A quantitative analysis of the performance of the proposed solution and further detail regarding the agent architecture is out-of-scope of this paper, which will focus on discuss the engineering aspects of the implementation of the proposed solution.

The testing scenario allowed to verify some benefits and drawbacks of the proposed approach. In this regard, it was verified that the asset can be easily integrated into the digital world through the AAS combined with the intelligence and collaborative capabilities provided by the agents. The only requirement is to describe the asset in a file-based AAS (e.g., using the AASX Package Explorer) and host that file on the AASX server (*Level 1*). In this case, a software solution was developed to generate automatically an agent according to the AASs hosted in the server, which simplify the deployment of agents as new AASs are added into the system.

Although there are several official standard submodel templates [24] provided by the Industrial Digital Twin Association (IDTA), the process of including all asset information in AAS submodels can be time-consuming, considering the variety of assets in a production system. In the future, suppliers are expected to provide the asset with its own AAS (at least a passive AAS). Moreover, the specificity of each application may require specific submodels, which requires extra efforts since they are not (yet) standardized, and can be a challenge to achieve interoperability.

The *Level 2* was demonstrated by implementing two modules (historical data and monitoring) that provide novel functionalities to the assets. The focus was not to present a complex implementation based on data analysis and AI techniques, but to present how modules can be included in the approach, in particular showing the idea of reusability, since a same module can be instantiated for different assets.

In addition, the testing scenario allowed to analyze how the PAs and RAs decentralize the intelligence and collaborate autonomously with each other to complete some production process (*Level 3*). In this regard, the system was able to handle the scalability, which was observed by increasing the number of assets (products) in the system, and therefore AASs and agents (mainly the interaction between the agents), and still complete all the designated processes.

In terms of reusability of the agents, the agent classes present a generic structure for all the different assets in the system, i.e., only one PA or RA class serves as the basis for all the products or resources. More specific functions can be added to the agent by including new behaviors, where each behavior has its own thread.

As a proof-of-concept, the agents were designed to collaborate following the FIPA Agent Communication specifications. However, by applying the agents in the I4.0, the interaction protocols (define the sequence of messages between the agents) and ontologies (define a common vocabulary and shared understanding among agents to promote the semantic interoperability) can be replaced/combined with the I4.0 language standards. This language defines the vocabulary and structure of messages (VDI/VDE 2193-1 standard) in a way that all I4.0 components understand each other. While the VDI/VDE 2193-2 standard defines the sequence of messages in a dialog of two or more I4.0 components.

Regarding the plug-and-produce capability of the proposed solution, tests were conducted to assess the seamless integration of new assets into the system. For this purpose, in an initial moment, it is considered the non-operation of one of the assets. However, at a given time during the process, the asset is introduced in the system, being available to produce the products. The performed tests in this scenario showed the capacity of a resource to become operational as soon as it is introduced into the system, without the need to stop, reconfigure and restart the system. Other tests were also performed, where resources were intentionally removed and the system remained stable, showing the flexibility of the approach to adapt to changes in the cell topology. This condition was achieved thanks the SOA principles combined with the agents, where each agent acts as service provider/consumer, by encapsulating the capability provided by the asset (e.g., punching, milling and drilling) and offering as services.

Finally, in terms of deployment and costs, all levels need to be implemented on computing platforms, considering the distribution across the Edge-Cloud. In this context, companies that do not have available resources to this end, can consider low-cost options, e.g., implementing *Level 1* in single-board computers (e.g., Raspberry Pi), as well as the *Levels 2 and 3*, as long as the applications do not require a significant processing power.

V. Conclusions and Future Work

This paper presented an approach to perform the digitization of production systems not fully positioned with the I4.0 technologies, architectures and standards, in order to be able to make this transition seamlessly by implementing different levels of digitization. These levels were implemented in a small-scale production system comprising several assets, using AASs to hold the asset information in a standard manner along the different lifecycle phases of the assets, external modules to provide novel functionalities to the assets, and MAS to provide mainly autonomy and collaborative capabilities to the assets.

Future work will be devoted to define the architecture of the agents, particularly considering the I4.0 language to provide the vocabulary, message structure and interaction protocols for the agents, as well as applying the proposed approach in different testing scenarios.

Acknowledgment

The authors are grateful to the Foundation for Science and Technology (FCT), Portugal, for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021). The author Lucas Sakurada thanks the FCT for the PhD Grant 2020.09234.BD.

References

[1] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Computers in Industry*, pp. 11–25, 2016.

[2] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial cyberphysical systems: A backbone of the fourth industrial revolution," *IEEE Industrial Electronics Magazine*, pp. 6–16, 2017.

[3] ZVEI and VDI/VDE, "Reference Architecture Model Industrie 4.0 (RAMI4.0)," 2015.

[4] Plattform Industrie 4.0, "Details of the Asset Administration Shell - from idea to implementation," 2019.

[5] Industrial Digital Twin Association, "Specification of the Asset Administration Shell - Part 1: Metamodel," 2023.

[6] ——, "Specification of the Asset Administration Shell - Part 2: Application Programming Interfaces," 2023.

[7] X. Ye, J. Jiang, C. Lee, N. Kim, M. Yu, and S. H. Hong, "Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach," *IEEE Industrial Electronics Magazine*, vol. 14, no. 4, pp. 146–157, 2020.

[8] X. Ye, S. H. Hong, W. S. Song, Y. C. Kim, and X. Zhang, "An Industry 4.0 Asset Administration Shell-Enabled Digital Solution for Robot-Based Manufacturing Systems," *IEEE Access*, vol. 9, pp. 154 448–154 459, 2021.

[9] K. Kanaan, J. Wermann, M. A. Bär, and A. W. Colombo, "Industry 4.0-compliant digitalization of a re-configurable and flexible laser cutter module within a digital factory," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT'23)*, 2023, pp. 1–7.

[10] S. Karnouskos, L. Ribeiro, P. Leitão, A. Lüder, and B. Vogel-Heuser, "Key Directions for Industrial Agent Based Cyber-Physical Production Systems," in *Proceedings of the IEEE International Conference on Industrial Cyber Physical Systems (ICPS'2019)*, 2019, pp. 17–22.

[11] B. Vogel-Heuser, M. Seitz, L. A. C. Salazar, F. Gehlhoff, A. Dogan, and A. Fay, "Multi-agent Systems to Enable Industry 4.0," *at - Automatisierungstechnik*, vol. 68, no. 6, pp. 445–458, 2020.

[12] L. Sakurada, P. Leitao, and F. De la Prieta, "Towards the Digitization using Asset Administration Shells," in *Proceedings of the 47th Annual Conference of the IEEE Industrial Electronics Society (IECON'21)*, 2021, pp. 1–6.

[13] P. Leitão, J. Barbosa, P. Vrba, P. Skobelev, A. Tsarev, and D. Kazanskaia, "Multi-agent System Approach for the Strategic Planning in Ramp-Up Production of Small Lots," in *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'13)*, 2013, pp. 4743–4748.

[14] M. Onori, N. Lohse, J. Barata, and C. Hanisch, "The IDEAS project: Plug & produce at shop-floor level," *Assembly Automation*, vol. 32, pp. 124–134, 04 2012.

[15] N. Antzoulatos, E. Castro, D. Scrimieri, and S. Ratchev, "A multi-agent architecture for plug and produce on an industrial assembly platform," *Production Engineering*, vol. 8, 12 2014.

[16] J. Queiroz, P. Leitão, J. Barbosa, E. Oliveira, and G. Garcia, "Agent-Based Distributed Data Analysis in Industrial Cyber-Physical Systems," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 3, no. 1, pp. 5–12, 2022.

[17] S. Karnouskos, P. Leitão, L. Ribeiro, and A. Colombo, "Industrial Agents as a Key Enabler for Realizing Industrial Cyber-Physical Systems: Multiagent Systems Entering Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 14, no. 3, pp. 18–32, 2020.

[18] IEEE, "IEEE Recommended Practice for Industrial Agents: Integration of Software Agents and Low-Level Automation Functions," *IEEE Std 2660.1-2020*, pp. 1–43, 2021.

[19] M. Jacoby, F. Volz, C. Weißenbacher, and J. Müller, "FA3ST Service – An Open Source Implementation of the Reactive Asset Administration Shell," in *Proc. of the IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA'2022)*, 2022, pp. 1–8.

[20] L. A. Cruz Salazar and B. Vogel-Heuser, "Comparison of agent oriented software methodologies to apply in cyber physical production systems," in *Proceedings of the IEEE 15th International Conference on Industrial Informatics (INDIN'17)*, 2017, pp. 65–71.

[21] L. A. Cruz Salazar and H. Li, *Proportional Reliability of Agent-Oriented Software Engineering for the Application of Cyber Physical Production Systems*. Cham: Springer International Publishing, 2018, pp. 139–156.

[22] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber–physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, 2016.

[23] Industrial Digital Twin Association, "Time Series Data," 2023.

[24] ——, "AAS Submodel Templates," 2023, Last accessed 28 June 2023. [Online]. Available: https://industrialdigitaltwin.org/en/content-hub/submodels