Engineering a Multi-agent Systems Approach for Realizing Collaborative Asset Administration Shells

Lucas Sakurada*, Paulo Leitao*, and Fernando De la Prieta†
* 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
† BISITE Digital Innovation Hub, University of Salamanca,
Edificio Multiusos I+D+i, 37007, Salamanca, Spain
Email: fer@usal.es

2022 IEEE International Conference on Industrial Technology (ICIT) | 978-1-7281-1948-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIT48603.2022.10002770

***Abstract*—In recent years, there has been a high demand for flexible and reconfigurable processes to meet the fast-changing market conditions. In this context, Industry 4.0 is promoting the digitization of traditional production systems towards intelligent factories with highly automated and rapidly adaptable capabilities based on Industrial Cyber-Physical Systems (ICPS). This digitization process is currently being leveraged by the so-called Asset Administration Shell (AAS), a standardized digital representation of an asset that provides the uniform access to the asset information. Additionally, the AASs offer the digital basis for future autonomous systems, where intelligent AASs may perform collaborative functions to enhance industrial processes. However, currently such solutions are still at an early stage of maturity. In this sense, this paper explores the adoption of a Multi-agent Systems (MAS) approach to provide the required intelligent and collaborative aspects for the traditional AAS. The proposed MAS-based AAS solution was applied in an industrial automation case study to analyze the feasibility of MAS to perform intelligent and collaborative functions in the AAS context.**

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

I. Introduction

The current trends for high-customized, low-cost and high-quality products impose strong requirements for production companies to adapt their systems towards flexible, reconfigurable and intelligent production systems based on Industrial Cyber-Physical Systems (ICPS) [1]. In this context, several national government strategies towards the digitization of the industry are promoting efforts to support this transition smoothly, e.g., “Industrie 4.0” in Germany, “Industrial Internet of Things” in USA and “Made in China 2025” in China [2]. All strategies are considered under the term Industry 4.0 (I4.0), sharing the same ground aiming to promote the digitization of production systems with intelligent, highly automated and rapidly adaptable capabilities combined with emergent technologies, namely Internet of Things (IoT), Artificial Intelligence (AI), Collaborative Robotics, Edge and Cloud Computing, and Virtual and Augmented Reality.

In such perspective, the Plattform Industrie 4.0 consortium relies on the Asset Administration Shell (AAS) [3] to leverage this digitization process. The AAS is a standardized digital representation of an asset (i.e., every logical or physical object that needs to be connected in the I4.0 network) that, together with its asset, form an I4.0 component as defined in the context of Reference Architecture Model Industrie 4.0 (RAMI4.0) [4]. Additionally, the AAS is perceived as an enabler to develop digital twins in a standardized manner, which provides uniform access and control of the asset’s information along its life-cycle, and enables interoperable communication among I4.0 components across the value-added network [3], [5].

The AAS presents a structure consisting of several submodels that hold the information and describe the functionalities of the asset in a standard manner, e.g., characteristics, properties, status, and capabilities [3]. Moreover, the AASs offer the digital basis for future autonomous systems, where intelligent AASs may perform collaborative functions to enhance industrial processes. However, the implementation practices are currently still restricted to passive and reactive AASs [6], i.e., the asset information is described in submodels, and it can be exchanged using a file, e.g., AASX (package file format for the AAS) or accessed via an Application Programming Interface (API). Further applications based on proactive AASs [6], which can make decisions, negotiate and interact autonomously with other AASs to achieve their goals, are still underexplored in the current state-of-the-art.

In this context, Multi-agent Systems (MAS) [7] are a suitable approach for decentralizing intelligence in I4.0 systems, being potential enablers to offer collaborative and intelligent capabilities for AASs. Derived from the area of Distributed Artificial Intelligence (DAI), MAS comprise a set of autonomous and cooperative entities, named agents, representing physical or logical objects in the system. In these systems, the global behaviour emerges from the interaction between agents through the implementation of interaction strategies, e.g., collaboration and negotiation, aiming to achieve the system goals aligned with the business purposes.

Having this in mind, this paper proposes an approach that integrates the three types of AASs (passive, reactive and proactive) into a single agent-based AAS solution, built up on the documentation provided by the Plattform Industrie 4.0 for developing AASs [8], [9], particularly passive and reactive solutions. Additionally, the main contribution of this paper resides in exploring the use of MAS to provide the intelligent and collaborative aspects required by a proactive AAS. The proposed agent-based AAS solution was applied in an industrial automation case study to analyze the feasibility of using MAS to enhance the collaborative functions in the AAS context.

The rest of the paper is organized as follows. Section II overviews the use of MAS in I4.0 and the existing related works for agent-based AASs solutions. Section III presents a method for developing collaborative and intelligent AASs, mainly using the MAS as a key enabler. Section IV describes the deployment of the proposed agent-based AAS solution in an industrial automation case study. Finally, Section V rounds up the paper with the conclusions and future work.

II. Multi-agent Systems in Industry 4.0

MAS have been advocated as a promising solution for emerging industrial challenges, particularly the industrial agents, which comply with several industrial requirements, namely hardware integration, reliability, fault-tolerance, scalability, industrial standard compliance, quality assurance, resilience, manageability and maintainability [10]. However, its wide adoption in industrial applications is still limited, particularly due to the complex and time-consuming task of developing customized solutions to meet the proprietary protocols offered by several automation vendors and creating the agents' local knowledge. Furthermore, the initial mistaken view that envisioned the agent to deliver the required control guarantees, replacing the existing system software, also contributed as one road blocker for the wider adoption of MAS in industrial environments. Today, it is recognized that the agents must be in charge of the so-called high-level control functions, acting in a different layer of the low-level control that is characterized by executing time-constrained functions.

In this context, the IEEE 2660.1-2020 standard has contributed to providing recommended practices for integrating intelligent software agents with low-level automation functions for a given application scenario [11]. In this viewpoint, the agent is responsible for the high-level control functions, most of the time operating in soft real-time. In turn, the physical asset's controller ensures the hard real-time execution of the agent's high-level decision. Complementary, the Plattform Industrie 4.0 relies on the AAS to provide a standardized description of the asset information, which may be a suitable solution to help create the agent's local knowledge in a standard way and thus ensure interoperability as also highlighted and preliminarily tested in [12].

In the literature, the agent-based AAS solutions are gaining more relevance in recent years and are considered a key direction for industrial agents. As suggested in [13], agents could be used to implement the AAS itself or provide key functionalities, e.g., data collection, collaborative functions, intelligent and autonomous decision-support. The work described in [14] investigates how the I4.0 emerging challenges in three different scenarios, namely Adaptable Factory, Order-Controlled Production and Self-organizing Adaptive Logistics, can be addressed with the support of MAS, mainly presenting that the agents can implement AASs. In [15], a discussion of how MAS can realize the AAS is presented, mapping their characteristics into AAS functionalities and indicating the possibility to use agents to extend the AAS functionalities embedding AI algorithms.

On the other hand, [16] proposes a pattern for the implementation of AASs based on industrial agents, showing the implementation of a resource AAS and a product AAS from the proposed pattern as proof of concept. A four-layered architecture and a methodology using industrial agents for the integration of AASs and physical assets is presented in [17], applying the proposed solution to integrate an agent-based AAS and a robotic arm. Lastly, [18] proposes an agent-based AAS approach for the development of AAS that considers agents to embed distributed intelligence and collaborative functions, holonic principles to provide the system organization and service orientation to support interoperability.

III. Development of Collaborative and Intelligent AASs

As aforementioned, the inclusion of intelligence and collaborative strategies to perform tasks are gaps in the current AAS state-of-the-art. In this context, this paper proposes a method for developing AASs using the MAS as a key enabler to distribute intelligence and offer collaborative functionalities. The proposed method takes advantage of the AAS specifications [8], [9], particularly the standardized information described in submodels and the exchange of information via API, to provide the required local knowledge for the agents, e.g., the functionalities offered by the asset along its lifecycle. This method considers the features of the different types of AAS [6], i.e., passive, reactive and proactive. For this purpose, three phases are proposed, namely *Passive Part Design*, *Reactive Part Design* and *Proactive Part Design*, to implement the main functionalities offered by each type of AAS.

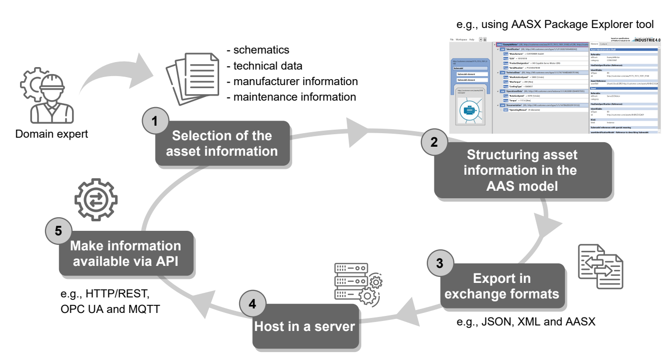

Figure 1. Steps to design the passive and reactive part.

A. *Passive Part Design*

The design of the passive part, as illustrated in Figure 1 (steps 1-3), follows the recommendations presented in [8] and consists of describing the asset's information and functionalities, e.g., characteristics, properties, status and capabilities in submodels. This information can be described in a standard manner by using the AASX Package Explorer tool (https://github.com/adminshell-io/aasx-package-explorer), a C# based viewer/editor for the AAS based on the specifications of the Plattform Industrie 4.0. The tool enables to export the asset's information in several file exchange formats, e.g., eXtensible Markup Language (XML), JavaScript Object Notation (JSON) and AASX. In this case, the AAS acts as a static file, where the asset's information is stored in a standard manner and can be exchanged digitally across the I4.0 network.

B. Reactive Part Design

The reactive part's design (see steps 4 and 5 from Figure 1) follows the recommendations presented in [9] and aims to enable the online access to the information provided by the AAS via API. For this purpose, such API needs to automatically parse the file-based AASs and provide the AASs information using different technologies, namely Hypertext Transfer Protocol (HTTP)/Representational State Transfer (REST), Open Platform Communications Unified Architecture (OPC UA) and Message Queuing Telemetry Transport (MQTT). Additionally, the design of the AAS reactive part can be extended to integrate the AAS to the asset, aiming to ensure the real-time operational data as illustrated in Figure 2. In this sense, it is necessary to develop an adapter, i.e., a software component that interfaces with the asset to collect data and updates the AAS submodels with this operational data via API.

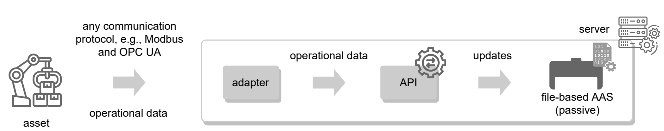

Figure 2. Reactive part to enable the integration between the AAS and asset.

C. Proactive Part Design

The design of the AAS proactive part focuses on realizing proactive AASs, i.e., decision-making entities that interact autonomously with other AASs to achieve their goals. In this context, the MAS provide intelligence and offer collaborative functionalities for the AASs. As illustrated in Figure 3, each set AAS-asset needs to have an associated agent that will use the information contained in the AAS submodels as local knowledge to know, e.g., its tasks, capabilities and restrictions. Based on that, the agents will be able to communicate by exchanging messages with each other, offering their asset's capabilities as services and looking for specific services to accomplish the manufacturing process of its asset. Moreover, it is important to highlight that it is necessary to specify how the agents will interact, e.g., using a request-response or publish-subscribe schema, and some specialization for the agents, e.g., resource and product agents. The creation of different types of agents increases the modularity and flexibility of the system since the agents can represent different types of assets and thus have distinct behaviours. In turn, the agents' behaviour depends on its role and goals and can be modelled using different formalisms, e.g., Unified Modeling Language (UML) (https://www.uml.org/) or Petri nets [19].

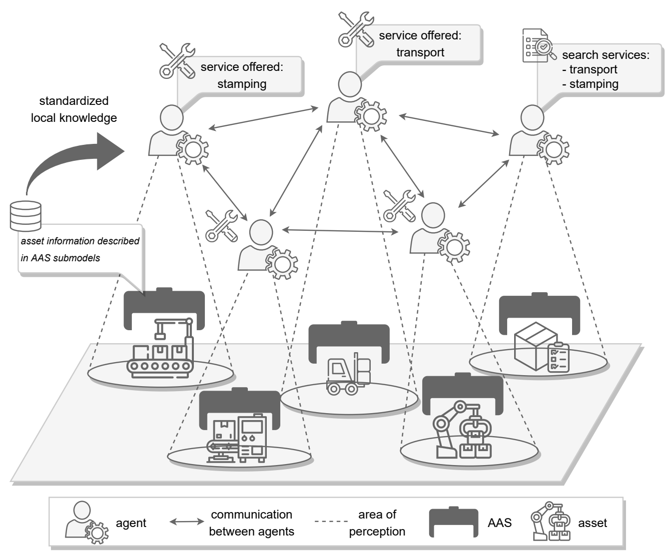

Figure 3. MAS as an enabler to distribute intelligence and offer collaborative functionalities to develop proactive AASs.

In addition, the agents can interact directly with the assets to request a control function using a specific communication protocol following the IEEE 2660.1-2020 standard [11] or indirectly by using the adapter concept. In this case, the adapter acts as a black box that interfaces with the asset using any communication protocol and offers a uniform way to interact with the asset, e.g., HTTP/REST.

IV. Deployment of the Agent-Based AAS Solution

The proposed solution was tested in an industrial automation case study, illustrated in Figure 4, developed using Factory I/O (https://factoryio.com/) and CODESYS (https://www.codesys.com/) software. The Factory I/O allows simulating assets (e.g., conveyors belts, sensors and pick-and-place robots), and complementary, the CODESYS enables controlling each asset's low-level functions through a virtual programmable logic controller (PLC) according to the IEC 61131-3 standard. Moreover, this setup offers the possibility to perform the interface with the assets using well-known industrial communication protocols, namely Modbus and OPC UA. The scenario comprises assets classified as resources and products, namely conveyor belts responsible for transporting the products, two machining centers to manufacture bases and lids from raw materials, a robot to assemble these parts and the product itself (combination of bases and lids to form the final product).

Based on the three phases mentioned in Section III, the *Passive Part* was developed using the AASX Package Explorer

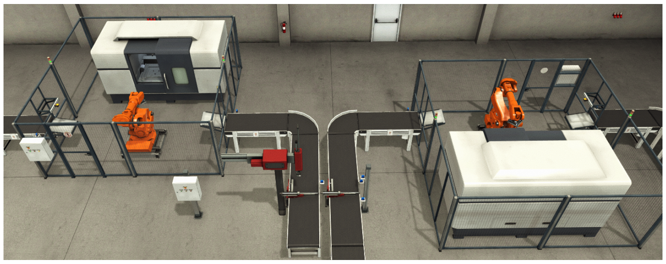

Figure 4. Industrial automation case study.

tool. For this purpose, a passive AAS (i.e., an AAS that only provides static information structured in submodels) was created for each asset presented in the testing scenario. It is important to highlight that the AAS must describe a wide range of information throughout the lifecycle of its asset. However, only information regarding the production process was considered in this work to facilitate the proof of concept. In this context, this approach contemplates some mandatory submodels, which provide the required knowledge for the agents to achieve their goals, namely "ManufacturingProcesses" and "Capabilities" submodels. The former provides information about the process that a product needs to go through, and the latter provides information about the capabilities offered by a resource. As an example, Figure 5 illustrates how the passive AAS for a product (top) and a resource (bottom) were developed using the AASX Package Explorer.

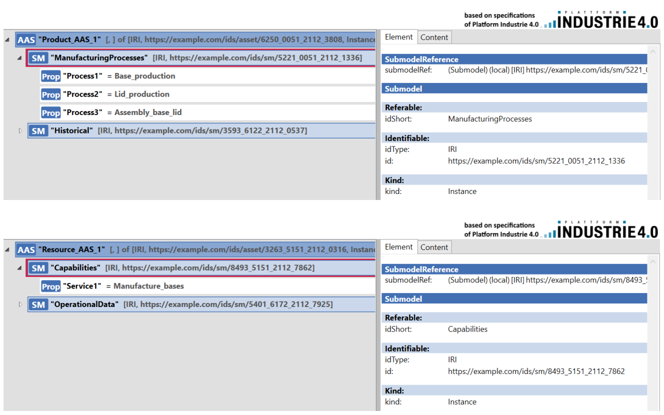

Figure 5. Example of passive AASs created with the support of the AASX Package Explorer. (Top) AAS for the product. (Bottom) AAS for the resource.

Aiming to develop the *Reactive Part*, the AASs created using the tool, i.e., for the two machining centers, the robot and the product, were exported as JSON files and hosted on a server. This server also hosts a RESTful API that automatically parses these files and provides the AASs information using REST, being possible to get information of the AAS, e.g., by executing a GET method `/aas/{aas-id}/submodels/ManufacturingProcesses` to obtain the information of the submodel "ManufacturingProcesses". This API was developed using Express.js (https://expressjs.com/), a framework for Node.js that provides minimal resources for building web servers. Moreover, the API enables the update of the static information of a passive AAS with real-time data from the asset. In this case, an adapter was implemented in Node-RED (https://nodered.org/) that interfaces with the asset using OPC UA to collect its operational data and thus updates the passive AAS via API using REST.

Regarding the development of the *Proactive Part*, the MAS were developed using the Java Agent DEvelopment (JADE) framework (https://jade.tilab.com/). JADE aims to simplify the development of MAS by providing a set of services and agents in compliance with the Foundation for Intelligent Physical Agents (FIPA) specifications, e.g., yellow-pages service, message transport and parsing service, and a library of FIPA interaction protocols. Moreover, JADE provides valuable 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) and resource agent (RA), according to the behaviour model illustrated in Figure 6. The RAs are associated with the AASs that represent assets classified as resources and similarly with PAs.

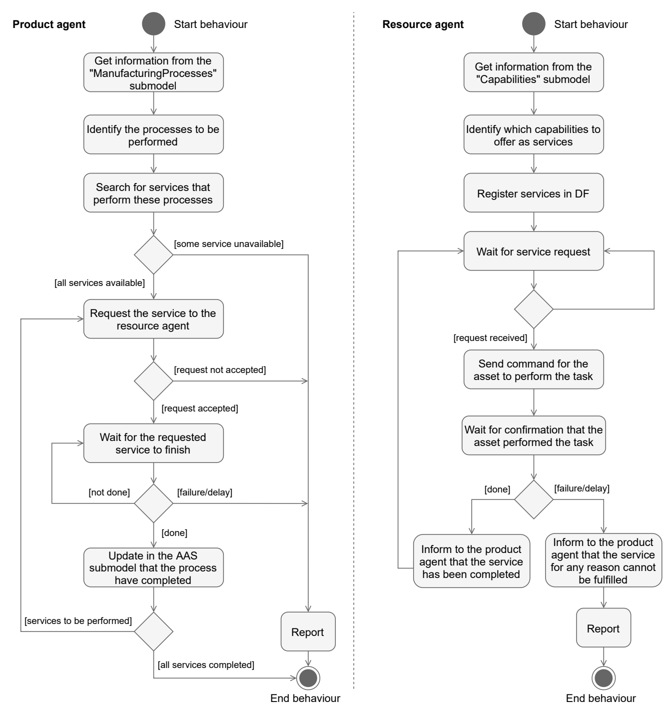

Figure 6. Behaviour model for the product and resource agents.

The RA uses the information described in the "Capabilities" submodel to register the functionalities of its associated asset as services in the Directory Facilitator (DF), i.e., the yellow pages service implemented by JADE. On the other hand, the PA search for a service in the DF to perform some specific process for its associated asset, described in the "ManufacturingProcesses" submodel. If the PA finds the required service, an interaction between the product and RAs

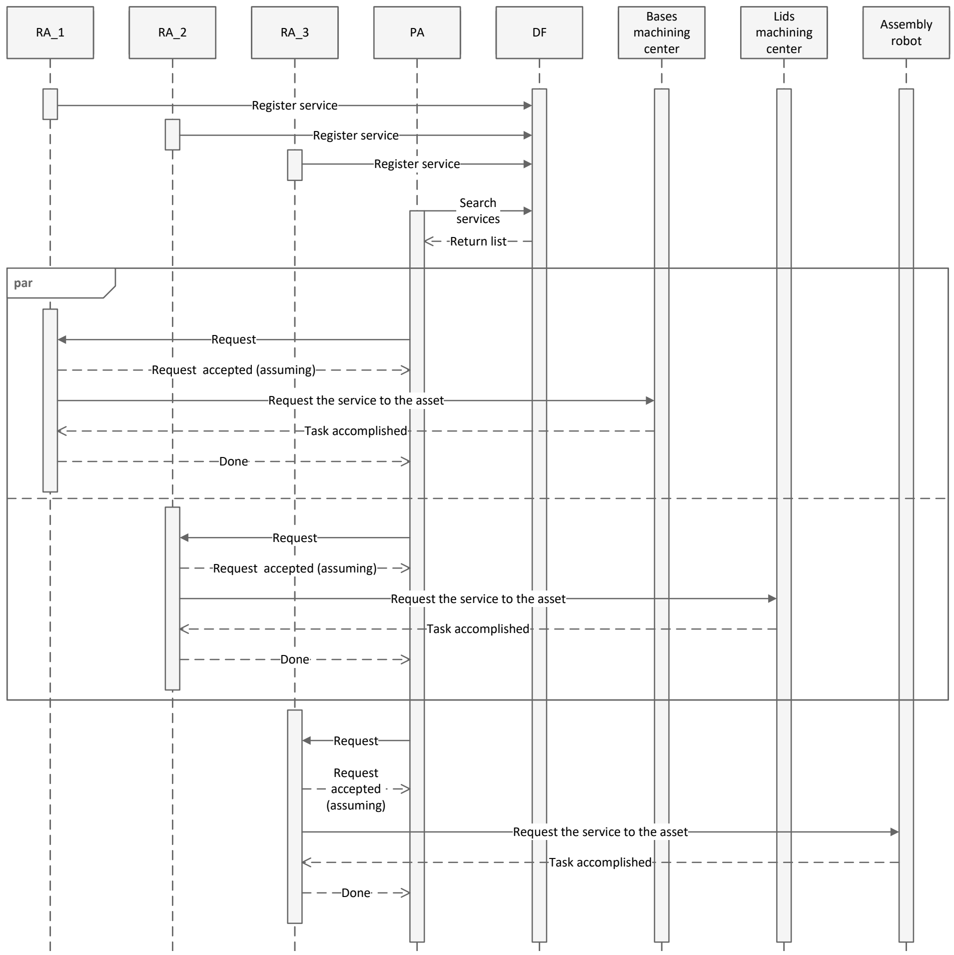

Figure 7. Procedures to registration and discovery of services, interaction between the agents and interaction with the physical assets.

is started following the FIPA Request Interaction protocol [20]. Moreover, considering that a RA accepts the request from the PA, the RA needs to interact with the physical asset. For this purpose, the "Capabilities" submodel also provides information of how to realize this interaction. In this case, a REST endpoint is informed for the RA to request the asset to perform the designated task. Finally, if some unexpected event occurs that stops the development of the process, e.g., a failure, the agents can report the occurred problem, e.g., sending an alert message to the operator.

Based on the implemented solution, an experimental test was performed to analyze the collaboration between the agents during the operational phase. In such a test, it was planned that the PA needs to interact with three RAs to accomplish the required processes for its asset. For this purpose, three required processes were included in the product's "ManufacturingProcesses" submodel. In contrast, each resource can provide one of these tasks, and this information was included in the "Capabilities" submodel of the resources. Both agents are created automatically according to the file-based AASs (JSON files) hosted on the server, and Table I summarizes the role of each agent in the system based on the AAS-asset sets.

Table I
AGENT ROLES BASED ON THE AAS-ASSET SETS.

Agent	Asset	AAS Submodel	Agent roles
RA_1	Bases machining center	Capabilities	Offer manufacture bases service
RA_2	Lids machining center	Capabilities	Offer manufacture lids service
RA_3	Assembly robot	Capabilities	Offer assembly service
PA	Product (base and lid)	Manufacturing Processes	Find and interact with agents that perform the following services: manufacture bases, manufacture lids and assembly

It is important to note that the conveyor belts do not have an associated agent, being considered part of the three resources used in the test, i.e., a resource always has a start and exit conveyor to transport a product between resources. As a preliminary result, Figure 7 illustrates the interactions performed between the PA and RAs, showing that the MAS can offer the required collaborative functionalities for the AAS. After each interaction with a RA, the PA updates this information in a "Historical" submodel of the associated AAS and starts the next interaction with the other RA. This submodel stores the steps that the product has already passed, providing a historical information of the asset along the process.

V. Conclusions and Future Work

The AAS assumes a crucial relevance in ICPS applications, acting as digital representation of industrial components, e.g., enabling to develop digital twins in a standardized manner. Its potential can be further explored by introducing intelligence and collaborative capabilities. Due to its inherent capabilities, MAS constitute a suitable approach to fulfil these requirements, being a key enabler for realizing proactive AASs.

This work was a preliminary step towards developing intelligent and autonomous AASs based on MAS, which is a research topic that has gained attention in the last few years. In this context, this paper presented a MAS approach to provide the intelligent and collaborative aspects required by a proactive AAS. This approach considers agents for realizing collaborative and coordinating tasks, where the agents represent assets (products or resources) and must interact with each other to achieve their goals. In addition, it was proposed an approach that integrates the three types of AASs (passive, reactive and proactive) into a single agent-based AAS solution. The proposed solution was applied in an industrial automation case study comprised of heterogeneous assets representing resources and products. In such system, the agents were created automatically based on the designed passive AASs for each asset in operation. According to the information contained in the AAS submodels, each agent was able to interact with the others to accomplish its tasks, showing the potentialities of the agent-based AAS for collaborative functions.

Future work will be devoted to performing more advanced experimental tests, particularly to analyze the system's performance in an environment comprised of a vast amount of assets, and therefore agents, as well as implementing other interaction strategies, e.g., following the FIPA Contract Net Interaction Protocol. Moreover, other behaviours can be embedded in the agents, aiming to introduce more intelligence capabilities to the proposed solution, e.g., to perform monitoring and diagnosis functions, which may be executed in parallel with the behaviours implemented in this work.

As a final remark, although the AAS metamodel allows describing the asset information in a standardized way, the designed submodels in this work to perform collaborative tasks, namely "ManufacturingProcesses" and "Capabilities", are not specified in any documentation, which may be a challenge to design agent-based AAS solutions in a standard manner. Therefore, future specifications or methodologies in this context are crucial to avoid misunderstanding among stakeholders. Moreover, the present paper adopted some solutions, e.g., Node.js, a simple and fast way to test the concept, being essential to consider converting the proposed application to a more industrial solution.

Acknowledgment

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UIDB/05757/2020. The author Lucas Sakurada thanks the FCT - Fundação para a Ciência e Tecnologia, Portugal, for the PhD Grant 2020.09234.BD.

References

[1] 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*, vol. 11, no. 1, pp. 6–16, 2017.

[2] Q. Li, Q. Tang, I. Chan, H. Wei, Y. Pu, H. Jiang, J. Li, and J. Zhou, "Smart manufacturing standardization: Architectures, reference models and standards framework," *Computers in Industry*, vol. 101, pp. 91–106, 2018.

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

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

[5] Industrial Internet Consortium and Plattform Industrie 4.0, "Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0 - An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper," 2020.

[6] Plattform Industrie 4.0, "Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment," 2021.

[7] M. Wooldridge, *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.

[8] Plattform Industrie 4.0 and ZVEI, "Details of the Asset Administration Shell - Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01)," 2020.

[9] ——, "Details of the Asset Administration Shell - Part 2 - Interoperability at Runtime - Exchanging Information via Application Programming Interfaces (Version 1.0RC01)," 2020.

[10] S. Karnouskos, P. Leitao, L. Ribeiro, and A. W. 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.

[11] P. Leitão, T. I. Strasser, S. Karnouskos, L. Ribeiro, J. Barbosa, and V. Huang, "Recommendation of Best Practices for Industrial Agent Systems based on the IEEE 2660.1 Standard," in *Proceedings of the 22nd IEEE International Conference on Industrial Technology (ICIT'2021)*, vol. 1, 2021, pp. 1157–1162.

[12] B. Vogel-Heuser, F. Ocker, and T. Scheuer, "An approach for leveraging Digital Twins in agent-based production systems," *at - Automatisierungstechnik*, vol. 69, no. 12, pp. 1026–1039, 2021.

[13] 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.

[14] 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.

[15] L. Sakurada and P. Leitão, "Multi-Agent Systems to Implement Industry 4.0 Components," in *Proceedings of the IEEE Conference on Industrial Cyberphysical Systems (ICPS'2020)*, vol. 1, 2020, pp. 21–26.

[16] A. López, O. Casquero, and M. Marcos, "Design patterns for the implementation of Industrial Agent-based AASs," in *Proceedings of the IEEE Conference on Industrial Cyberphysical Systems (ICPS'2021)*, 2021, pp. 213–218.

[17] A. López, O. Casquero, E. Estévez, P. Leitão, and M. Marcos, "Towards the generic integration of agent-based AASs and Physical Assets: a four-layered architecture approach," in *Proc. of the IEEE 19th International Conference on Industrial Informatics (INDIN'2021)*, 2021, pp. 1–6.

[18] L. Sakurada, P. Leitao, and F. De la Prieta, "Agent-Based Asset Administration Shell Approach for Digitizing Industrial Asset," in *Proceedings of the 14th IFAC Workshop on Intelligent Manufacturing Systems (IMS'2022)*, 2022.

[19] T. Murata, "Petri nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[20] FIPA, "FIPA Request Interaction Protocol Specification." [Online]. Available: http://www.fipa.org/specs/fipa00026/index.html