Recommendation of Best Practices for Industrial Agent Systems based on the IEEE 2660.1 Standard

Paulo Leitão*, Thomas I. Strasser†, Stamatis Karnouskos‡, Luis Ribeiro§, José Barbosa¶, Victor Huang‖

*Research Center in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal. Email: pleitao@ieee.org

†AIT Austrian Institute of Technology and TU Wien, Vienna, Austria. Email: thomas.i.strasser@ieee.org

‡SAP, Walldorf, Germany. Email: karnouskos@ieee.org

§Linköping University, SE-58 183 Linköping, Sweden. Email: luis.ribeiro@liu.se

¶MORE Colab, Avenida Cidade de Léon, 506, 5301-358 Bragança, Portugal. Email: jbarbosa@ieee.org

‖Sage Technology Resources, CA, USA. Email: v.huang@ieee.org

***Abstract*—Cyber-Physical Systems (CPS) is a key concept in Industry 4.0, acting as a backbone to develop smart processes, machines and products. Multi-Agent Systems (MAS) is a suitable paradigm for the realization of such industrial CPS systems, supporting the distribution of intelligence and decision-making capabilities among a network of autonomous and cooperative agents. Standardization is a key factor for the acceptance of industrial CPS and agent-based systems, assuming a critical role in establishing specifications for the hardware integration, which is an important requirement for industrial agents. This paper focuses on the recently established IEEE 2660.1-2020 standard that defines a recommended practice to solve the interface problem when applying industrial agents, namely integrating intelligent software agents with low-level automation devices in the CPS context. The paper illustrates the applicability of the standard in three different application scenarios related to power and energy systems, factory automation and building automation, and discusses future directions in terms of standardization in the field of industrial agents, that are required for its wider adoption in the realization of industrial CPS solutions.**

I. Introduction

Cyber-Physical Systems (CPS) is a key concept in Industry 4.0 (I4.0), acting as a backbone to develop emergent production systems and contributing to the digital transformation of production processes towards smart processes, machines and products. CPS focuses on the integration of computational applications with physical assets, being designed as a network of interacting cyber and physical counterparts to form a large system [1]. This cyber-physical perspective recalls the holonic concept and particularly the Asset Administration Shell (AAS) [2] that is one specification of the Reference Architecture Model for I4.0 (RAMI4.0) [3]. AAS is the digital representation of assets that together constitutes the I4.0 components, allowing the access and control of the asset's information and provides an interface communication with other I4.0 devices based on the service-oriented architecture [2], [4]–[6].

Multi-agent systems (MAS) [7] is a suitable technology, derived from the distributed artificial intelligence field, to implement flexible, adaptive and responsive industrial CPS, based on a set of intelligent autonomous entities, called agents, that cooperate to achieve the system goals. MAS provides an alternative approach to design these systems by distributing the intelligence, where the decision/control is not centered in a centralized node but instead emerge from the interaction among the distributed autonomous agents. Industrial agents inherit the foundations of software agents, namely the intelligence, autonomy, and cooperation principles, but they are designed to operate in industrial environments and applications and, therefore, need to adhere to industrial requirements, e.g., hardware integration, reliability, fault-tolerance, scalability, industrial standard compliance, quality assurance, resilience, manageability, and maintainability [8]–[10].

An important requirement in industrial agents is the integration of the physical (hardware) assets, where the agents implement the cyber counterpart in a CPS context. One key example that represents the challenges for industrial agents is the effort to realize the interface between the software agents, which implement the AAS functions enhanced with intelligence capabilities, and the assets, which implement the low-level automation function, as illustrated in Figure 1. Due to the heterogeneity of the counterparts, this interface can be implemented in different ways, without having a unique standard that allows the easy, fast and transparent integration.

Having this in mind, the recent established IEEE 2660.1-2020 Recommended Practice on Industrial Agents [11] defines

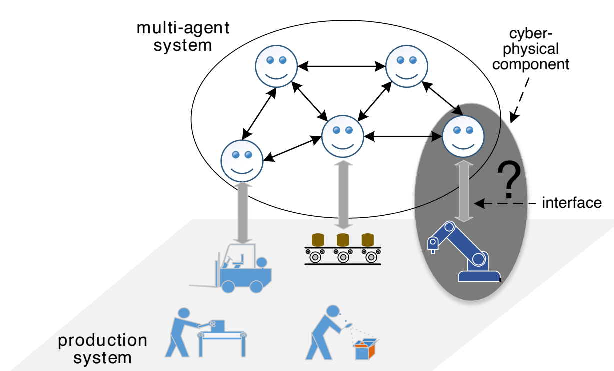

Figure 1. Interface problem in industrial agents.

a method to recommend the best interfacing practices for a particular application scenario taking into account the feedback from experts in implementing and using different interfacing practices.

IEEE 2660.1-2020 is a new approach in the area, something that has not been realized for several years now, when one considers that the interest in software agent standardization peaked almost two decades ago, e.g., with the Foundation for Intelligent Physical Agents (FIPA) [12]. FIPA has produced specifications for the development of heterogeneous agent-oriented software solutions; in particular, establishing the basis for the structure of the agents and their intercommunication, but lacking the integration of software agents and physical assets. However, the accomplishment of industrial environments imposes specific requirements that are not covered by FIPA specifications like the hardware integration, i.e., the interconnection of intelligent software agents with hardware devices performing (real-time) control.

Industrial CPS (ICPS) have emerged as a new focus area, and the excellent matching between ICPS challenges and industrial agents [1], [13]–[16] poses a new opportunity for the agent community to address ICPS needs with them [17]. Such an undertaking, however, will need up to date standards, that are at par with ICPS, and address beyond several factors that are related to the acceptance of ICPS and agent-based systems [10]. This paper relates to some of these aspects, as it focuses on the IEEE 2660.1-2020 and its main principles for following the recommended practice. It also illustrates the utilization of IEEE 2660.1 in three different application scenarios, as well as future directions that are necessary to be taken, in terms of standardization in the field of industrial agents.

The rest of the paper is organized as follows. Section II overviews the main principles of the recommended practice and Section III describes its application to three different application scenarios derived from power and energy systems, factory automation and building automation areas. Section IV discusses possible standard initiatives beyond IEEE 2660.1-2020 in the field of industrial agents. Finally, Section V rounds up the paper with the conclusions and points out future work.

II. Recommendation Practice Overview

The recommended practice IEEE 2660.1-2020 establishes a method to recommend the best interface practice to integrate a software agent and a physical asset performing low-level automation functions, taking into consideration the feedback of experts in using different interface practices. This method contributes to leverage the best practices of developing industrial agents for specific automation control problems in the industrial CPS context, improving the reuse, consistency and transparency in the integration of industrial agents and low-level control functions.

A. Scope of Application

The IEEE 2606.1-2020 standard answers the question related to the best practices to implement the interface between a software agent and the physical asset, i.e., hardware controller, considering a particular application scenario defined by the user. The standard strongly relies on the collection of feedback provided by experts in the field in implementing and using different interfaces practices, which their proper assessment can be used to match the best-scored ones for a particular application scenario.

For this purpose, it recommends the best interfacing practice by providing information regarding the interface configuration and technologies to be considered. As examples, the approach provides information about the best interface practices to interconnect software agents with a sensor (e.g., for monitoring the acquired measure), a robot controller (e.g., for control functions, namely to start/stop the execution of a program), and a Programmable Logic Controller (PLC) that control a conveyor system (e.g., to read and write variables), as illustrated in Figure 2.

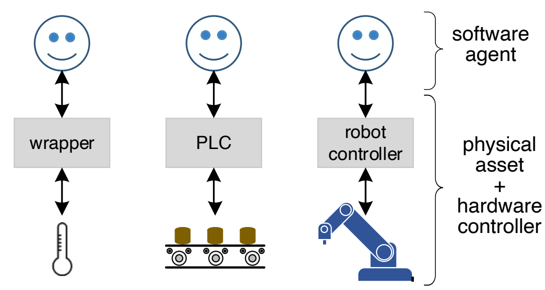

Figure 2. Examples of physical assets to be interconnected by software agents.

Other issues like the definition of criteria and metrics to perform the assessment of existing interface practices are not considered in this standard, as well as providing information regarding the performance of each interface practice (e.g., the response time).

B. Interface Patterns

Several interface practices to interconnect the software agents and the physical assets are reported in the literature, using different approaches and technologies [18]. These different types of interface practices are clustered according to three dimensions, namely the location, the interaction mode and the used technology (see Figure 3).

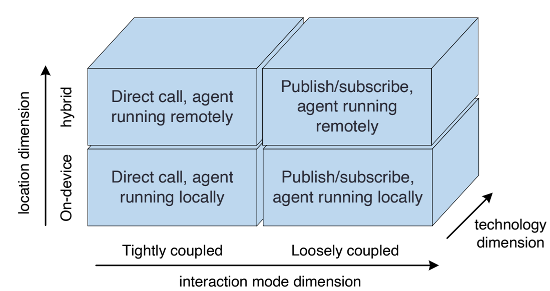

Figure 3. Types of interfacing patterns (adapted from [19]).

The location dimension refers to the place where the software agent is running, i.e., if it is running in the same computational platform of the physical asset or in a remote one. The interaction mode dimension refers to the way the interaction is performed, e.g., following a direct call under the client-server approach or using a publish-subscribe approach that ensures interoperability. The third dimension is related to the different technological options to implement the interface, e.g., considering proprietary technologies or more standardized ones such as OPC-UA and Modbus. Of course, each interface practice presents strengths and weaknesses that correspond to different quality indicators.

C. Recommendation Method

Having different ways to implement the interface, the standard provides a method to recommend the best practice for a specific scenario, scoring and comparing the existing interface practices according to specific criteria, and also indicating their strong and weak points. For this purpose, the methodology to recommend the best interface practice comprises the scoring engine and the recommendation engine.

IEEE 2606.1-2020 strongly relies upon the feedback of experts with respect to the assessment of the several interface practices according to their experience. In this context, experts can continuously update the repository of interface practices, assessing each one according to a set of criteria that highlight their best features and the scoring engine supports the user in the assessment of them. As an example, the ISO/IEC 25010 standard [20], that proposes quality characteristics to be considered for the evaluation of a software product, can be used to assess the interface practices. ISO/IEC 25010 also defines sub-characteristics for each characteristic, as illustrated in Figure 4, that can be measurable according to the ISO/IEC 25023 standard [21] specifications. Examples of quality criteria used in this standard are the response time, scalability and re-usability parameters.

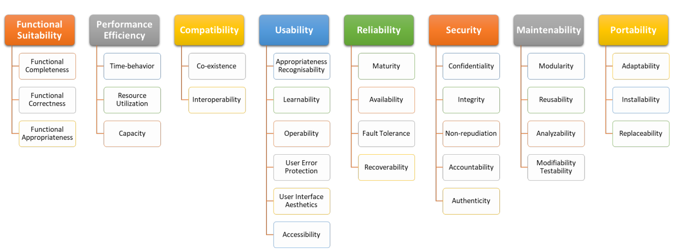

Figure 4. Overview of ISO/IEC 25010 product quality model characteristics (adapted from [20]).

In such way, an expert, when providing feedback related to her/his expertise in implementing and using a certain interface practice, makes an assessment of the response time, scalability and re-usability characteristics using a 5-level Likert scale. Since several experts may provide feedback for similar setups, several entries for the same technological interface practice may be found in the repository. Therefore it is necessary to perform aggregation of the different entries for each practice, e.g., using the statistical average of the different scoring values for each criterion characterizing the interface practice.

The recommendation engine analyzes the scored agent interface practices that are hosted in the repository (previously assessed by experts), aiming to select the one that better fits the application scenario context defined by the user, which includes the function, application domain, and technological constraints. For this purpose, the first step is to remove the interface practices that do not fulfill the context and technological constraints defined by the user. As an example, if hardware controllers are not able to host agents, then all on-device practices are not considered. The remaining practices are assessed by applying an appropriate simple or complex recommendation algorithm that, e.g., can use a multi-criteria function that weights each criterion according to its level of relevance for the application scenario. As an example, the user can define that for a specific application scenario, the response time is a critical characteristic and should be weighted with 80%, and the scalability is less relevant but also important to be considered, and should be weighted with 20%, summing 100%. As output, the engine provides the recommended interface practice and a list of alternatives, sorted according to the scoring of the selected recommendation algorithm.

III. Application of the IEEE 2660.1 Method

The application of the recommended practice is illustrated with three examples from different fields, namely power and energy systems, factory automation, and building automation. In this exercise, a repository of 23 assessed interface practices by experts was considered, covering the different interface patterns illustrated in Figure 3, using different technologies.

A. Power and Energy Systems

The usage of agent-based technology in the domain of power and energy systems is manifold [22]; it can reach from simple monitoring and control actions to quite complex diagnostics and (self-)reconfiguration tasks.

Simple control actions can be the ON/OFF switching of relays for powering streetlights and corresponding signs along a street for energy savings. The involved devices cannot host the agents due to resource constraints (i.e., only the control code for the switching actions are executed on these devices), and the software agents are usually executed remotely on a more powerful computing environment.

A more complex and typical application scenario for power and energy systems is the control of a microgrid. In such a setup distributed, renewable energy resources (photovoltaic generators, small wind turbines, etc.), diesel generators, battery energy systems, and loads are locally connected together via power lines for off-grid applications, but also a connection to the main power network is possible in case of a grid-connected solution. The use case of such a scenario is to manage the balance of the local generation and consumption, and this is carried out in an auction-based way by a distributed, agent-based control [22]. Typically, the involved components have enough computing power to host and execute software agents. The time behavior/response of such a distributed control application is the most important factor in such a configuration, but also scalability and re-usability are essential points. Such criteria (weighted as: time response 60%, scalability 30%, re-usability 10%) applied to the IEEE 2660.1-2020 selection method for this particular scenario produces the results illustrated in Figure 5.

Results						
Recomended interface practice						
Id practice	Location	Interaction mode	API client	Channel	Broker	Score
OT-1	On-device	Tightly coupled	Java			2.35
Details						
Id practice	Location	Interaction mode	API client	Channel	Broker	Score
OT-1	On-device	Tightly coupled	Java			2.35
HT-7	Hybrid	Tightly coupled	Java	OPC UA		1.92
OL-1	On-device	Loosed coupled	REST/JSON	MQTT	Eclipse Mosquito	1.92
HT-5	Hybrid	Tightly coupled	Apacho Milo	OPC-UA		1.78
OT-3	On-device	Tightly coupled	C/C++/SQL	DBMS		1.78
HT-3	Hybrid	Tightly coupled	Java	DPWS		1.78
HL-2	Hybrid	Loosed coupled	Java	OPC-UA		1.65
OT-2	On-device	Tightly coupled	C/C++	Sockets		1.49
HT-2	Hybrid	Tightly coupled	Java	Sockets		1.44
HT-4	Hybrid	Tightly coupled	C/C++/SQL	Sockets		1.37
HT-6	Hybrid	Tightly coupled	Java	Etherner/IP		1.34

Figure 5. Results of the recommended practice for power and energy systems.

Due to the possibility of embedding agents directly into the aforementioned power units with embedded controller, the recommended practice is a tightly coupled on-device approach, whereas the low-level automation functions and the agent-based optimization algorithms are executed on the same platform. However, also other coupling possibilities (e.g., hybrid, tightly-coupled or on-device, loosely-coupled) are feasible, as shown in the results above.

B. Factory Automation

A typical case of industrial agents in a factory automation domain includes using the agent to take care of complex soft real-time operations, such as planning, scheduling or system self-organization under disturbances while traditional industrial controllers, particularly PLCs, implement the hard real-time control strategies that maintain the nominal operation of the system. Most PLCs do not allow agent code, which is typically developed in JAVA, C#, C++, or Python, to run natively.

In such a scenario, the agents need to ideally communicate with the PLC at constant intervals, normally such intervals will be longer than the PLC cycle time, and the actual interaction over the network may include several PLC cycles. For example, the agent may communicate with the PLC every 60 ms (soft real-time), and the PLC will have a cycle time of 5 ms (hard real-time). Processing of the agent communication may take 2 PLC cycles before it becomes effective in the PLC. In such conditions, a user would value time response as the most critical criterion since there are only so many deadlines that can be missed without consequence for the system. Scalability is important but to a lesser extent in the specific case. This, for instance, would result in weighting time response at 80% and scalability at 20%. Such criteria applied to the IEEE 2660.1-2020 selection method for this application scenario produces the results in Figure 6.

Results						
Recomended interface practice						
Id practice	Location	Interaction mode	API client	Channel	Broker	Score
HT-7	Hybrid	Tightly coupled	Java	OPC UA		3.20
Details						
Id practice	Location	Interaction mode	API client	Channel	Broker	Score
HT-7	Hybrid	Tightly coupled	Java	OPC UA		3.20
HT-5	Hybrid	Tightly coupled	Apache Milo	OPC-UA		3.04
HT-3	Hybrid	Tightly coupled	Java	DPWS		2.72
HT-6	Hybrid	Tightly coupled	Java	Ethernet/IP		2.24
HT-2	Hybrid	Tightly coupled	Java	Sockets		1.85
HT-4	Hybrid	Tightly coupled	C/C++/SQL	Sockets		1.63
HT-1	Hybrid	Tightly coupled	Java	Modbus		1.58
HL-2	Hybrid	Loosely coupled	Java	OPC-UA		1.45
HL-1	Hybrid	Loosely coupled	Apache Paho	MQTT	Eclipse Mosquitto	1.44
HL-3	Hybrid	Loosely coupled	C/C++/SQL	Sockets	DBMS	1.15
HL-4	Hybrid	Loosely coupled	Apache Paho	MQTT		0.41

Figure 6. Results of the recommended practice for factory automation.

The suggested practice is a tightly coupled interface using OPC-UA, but with a relatively low score. This would suggest that none of the practices would be a very good fit for the scenario described. However, a closer look at the actual practices reveals that many of the suggested practices could work if the appropriate computational platforms are used. This suggests that a better strategy is required to characterize the time response of the different practices.

C. Building Automation

Building automation, particularly when applied for large services/offices buildings, assumes an important field of application for industrial agents. In fact, a multitude of automation functions are already present in many buildings, and this number is expected to grow over the years, mainly through the application of the Internet of Things (IoT) technologies.

The considered scenario assumes a large office and scientific laboratories building used for established companies, start-ups, and research and innovation laboratories. Many features are expected from such a building offering, particularly environmental comfort, security insurance, and energy savings. The building has installed a set of automated functions and a network of distributed sensor information, being this information made available throughout a dedicated network running on an RS-485 communication bus and on a KNX bus.

Being an office building, it is not expected, from the industrial agent perspective, high requirements regarding responsiveness. Heating, Ventilation and Air Conditioning (HVAC) and air renewal control have slow time constants, as environmental parameters evolve in the time-frame from minutes to hours. Regarding monitoring, the same assumptions can be stated due to the same time-frame. On the other side, in terms of scalability, this assumes a critical feature, introducing elasticity to the introduction and removal of devices in the network. Therefore, one could assume a 15% weight for the response time and 85% for the scalability. Such criteria applied to the IEEE 2660.1-2020 selection method for this application scenario produces the results illustrated in Figure 7.

The suggested practice has a high evaluation, 4.7 over 5, and follows a loosely coupled approach with MQTT (Message Queuing Telemetry Transport). In fact, if scalability is

Results						
Recomended interface practice						
Id practice	**Location**	**Interaction mode**	**API client**	**Channel**	**Broker**	**Score**
HL-4	**Hybrid**	**Loosely coupled**	**Apache Paho**	**MQTT**		**4.70**
Details						
Id practice	**Location**	**Interaction mode**	**API client**	**Channel**	**Broker**	**Score**
HL-4	Hybrid	Loosely coupled	Apache Paho	MQTT		4.70
HT-4	Hybrid	Tightly coupled	C/C++/SQL	Sockets		3.76
HL-3	Hybrid	Loosely coupled	C/C++/SQL	Sockets	DBMS	2.96
HT-2	Hybrid	Tightly coupled	Java	Sockets		2.73
HT-7	Hybrid	Tightly coupled	Java	OPC UA		2.56
HL-2	Hybrid	Loosely coupled	Java	OPC-UA		2.47
HL-1	Hybrid	Loosely coupled	Apache Paho	MQTT	Eclipse Mosquitto	2.40
HT-5	Hybrid	Tightly coupled	Apache Milo	OPC-UA		1.89
HT-3	Hybrid	Tightly coupled	Java	DPWS		1.50
HT-6	Hybrid	Tightly coupled	Java	Ethernet/IP		1.29
HT-1	Hybrid	Tightly coupled	Java	Modbus		0.98

Figure 7. Results of the recommended practice for building automation.

more demanded than responsiveness, a loosely-coupled, e.g., a publish/subscribe approach, would be recommended as the architectural choice for industrial agent interaction. Industrial agents typically would be more devoted to monitoring functions but, as algorithms and people's confidence grow in such technologies, it is expected that industrial agents, gradually, assume also control functions, increasing environment comfort while decreasing the overall energy consumption.

IV. Beyond IEEE 2660.1-2020

New, as well as extensions to existing standards relevant to industrial agents, are needed to efficiently integrate agents in the ICPS domain [17]. While IEEE 2660.1-2020 provides some guidelines for the best interfacing practices, there are still several issues that need to be addressed. Some motivation and potentially prioritization towards addressing these issues can be found by having a closer look at the common area defined by the ICPS needs and priorities [1], [13]–[15], [17] with the key factors that lead to the industrial acceptance of agent-based solutions [10]. This section discusses possible future standards related to the IEEE 2660 series.

A. Beyond FIPA Towards Industrial Requirements

One potential area of future work constitutes the area already defined by several FIPA standards [12]. While this standards family defines well the structure of agent-based solutions and the interactions among agents, they have not been updated for several years, and consequently, there is no strong link to the modern visions of integrating agents in ICPS. It, therefore, seems that a closer re-inspection of the FIPA standards, and their potential extension to accommodate the needs of ICPS within the larger context of architectural and communication patterns envisioned in the I4.0 context, is necessary.

B. Benchmarking and Metrics

An important issue in IEEE 2660.1-2020 is the criteria used to assess the existing interfacing practices, allowing to score and compare them in order to recommend the best practice for a particular application scenario. At the moment, the standard considers a set of criteria derived from the ISO/IEC 25010 standard for the assessment of software applications [23], [24]. However, the specific requirements of the interface practices require the definition of additional and more customized assessment criteria that are not covered and provided by ISO/IEC 25010, in order to fully enable technical comparison like on performance characteristic [25]. With these in mind, significant efforts should be provided in the definition of proper criteria for the assessment of existing interface practices, taking into consideration the particularities of the interface between software agents and physical assets.

C. Time Critical Applications and Standards Harmonization

As digital transformation gains acceptance in ICPS, defining the role of MAS and industrial agents become increasingly critical in its key systems requirements for distributed industrial automation environments. As ICPS systems embrace a variety of standards for strategic and best practices, industrial agents need to be introduced and harmonized with these other standards for seamless integration and enhanced systems performance, interoperability, and maintenance. A possible area of focus can be time-critical applications in providing industrial agents to low latency sensitive areas such as edge computing nodes and platforms in industrial automation and time-sensitive sensor industrial networks [26]. The time-critical aspects of several industrial applications need to be properly considered when designing, developing, and operating agent-based industrial solutions.

D. Artificial Intelligence and Ethics

The latest practical advances in Artificial Intelligence (AI) have enabled significant benefits in the implementation of monitoring, diagnosis, prediction, optimization, and planning tasks, in industrial agents context. Overall, this has an impact as the learning capabilities of agents are impacted and can move beyond traditional ways of learning or inferring, which can nowadays be also done in an effective way on the edge. This implies that it is possible for individual agents, as well as swarms, to capitalize on collaborative learning at the system level, and thereby also utilize distributed resources and capabilities like at edge, fog and cloud [27].

Such practical applications that go beyond the strictly controlled environments (e.g., of a shop-floor into real-world applications in uncertain environments), also raise several additional challenges beyond the traditional security and safety ones [28]. For instance, issues related to ethics, bias, compliance to regulatory frameworks, culture relevance, etc., need to be addressed. In addition, such issues need to be also integrated in the criteria considered as a best practice that is discussed in this work, which currently is technology-focused and does not sufficiently include these additional aspects. Therefore, it could be that the mechanically best-proposed practice is not appropriate in specific contexts (culture), or not fully compliant to local laws, etc.

In terms of predictability and deterministic behavior that can be designed, prototyped, evaluated, and certified, there are several challenges that need to be addressed. Such envisioned systems that depend on stohastically-behaving components (e.g., agent-based behavior which due to learning may change over time), need to adhere to concrete requirements for performance, compliance, ethic norms etc., in a consistent way, that can subsequently be certified and utilized in society. Even if this is achieved, it has also to be considered that such systems that evolve over time contribute to emergent behaviors, which pose another level of complexity and issues that span several technologies and practices.

Furthermore, all systems, including agents, that learn and evolve must be aware of the risk related to bias in the data they learn from; biased training data will lead to biased inference, which in real-world approaches can lead e.g., to discrimination of humans they interact with. Therefore, it becomes evident that such considerations need to be addressed in a standardized and holistic way for engineers to be able to design systems that integrate cutting-edge technologies, which adhere to societal requirements.

V. Conclusions and Future Work

MAS is a suitable paradigm for the realization of flexible production systems within the context of ICPS, providing key capabilities such as distributing intelligence and decision-making capabilities among a network of distributed autonomous and cooperative agents. In industrial agents, the interconnection between the software agent and the physical asset, including the hardware controller, assumes a crucial role.

This paper describes the IEEE 2660.1-2020 standard that defines the recommended practices to solve the interface problem when applying industrial agents, namely integrating intelligent software agents with low-level automation devices in the context of ICPS. To illustrate its applicability, three examples of application in power and energy systems, factory automation, and building automation are provided. This standard strongly contributes to supporting engineers and software developers by leveraging the best practices of developing industrial agents, improving the reuse, consistency, and transparency in the integration of industrial agents and low-level automation control functions.

The paper also outlines future directions in terms of standardization in the field of industrial agents that are necessary to be taken for its adoption in the realization of ICPS solutions, i.e., aspects related to benchmarking and metrics, time critical applications, artificial intelligence, and ethics.

References

[1] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Comp. in Industry*, vol. 81, pp. 11–25, Sep. 2016.

[2] ZVEI, "Specification - details of the asset administration shell," 2019.

[3] ——, "Reference architecture model industrie 4.0 (rami4.0)," 2015.

[4] X. Ye and S. H. Hong, "Toward industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Industrial Electronics Magazine*, vol. 13, no. 1, pp. 13–25, Mar. 2019.

[5] J. Fuchs, J. Schmidt, J. Franke *et al.*, "I4.0-compliant integration of assets utilizing the asset administration shell," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2019.

[6] G. Aceto, V. Persico, and A. Pescape, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3467–3501, 2019.

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

[8] P. Göhner, Ed., *Agentensysteme in der Automatisierungstechnik*. Springer Berlin Heidelberg, 2013.

[9] R. Unland, "Industrial agents," in *Industrial Agents: Emerging Applications of Software Agents in Industry*, P. Leitão and S. Karnouskos, Eds. Elsevier, 2015, pp. 23–44.

[10] S. Karnouskos and P. Leitao, "Key contributing factors to the acceptance of agents in industrial environments," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 696–703, Apr. 2017.

[11] *IEEE 2660.1 Recommended Practices on Industrial Agents: Integration of Software Agents and Low Level Automation Functions*, IEEE Std., 2020. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=9153208

[12] "Foundation for Intelligent Physical Agents (FIPA)," IEEE Computer Society. [Online]. Available: http://www.fipa.org

[13] P. Leitao, S. Karnouskos, L. Ribeiro *et al.*, "Smart agents in industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.

[14] A. W. Colombo, S. Karnouskos, O. Kaynak *et al.*, "Industrial cyber-physical systems: A backbone of the fourth industrial revolution," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 6–16, Mar. 2017.

[15] S. Karnouskos, L. Ribeiro, P. Leitao *et al.*, "Key directions for industrial agent based cyber-physical production systems," in *2019 IEEE Intern. Conference on Industrial Cyber Physical Systems (ICPS)*, May 2019.

[16] P. Leitao, F. Pires, S. Karnouskos *et al.*, "Quo vadis industry 4.0? position, trends, and challenges," *IEEE Open Journal of the Industrial Electronics Society*, vol. 1, pp. 298–310, 2020.

[17] S. Karnouskos, P. Leitao, L. Ribeiro *et al.*, "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, Sep. 2020.

[18] P. Leitao, S. Karnouskos, L. Ribeiro *et al.*, "Common practices for integrating industrial agents and low level automation functions," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2017.

[19] ——, "Integration patterns for interfacing software agents with industrial automation systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2018.

[20] *ISO/IEC 25010 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, ISO/IEC Std. 25 010 Std., 2011. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en

[21] *ISO/IEC 25023 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality*, ISO/IEC Std. 25 023 Std., 2016. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:25023:ed-1:v1:en

[22] P. Vrba, V. Marik, P. Siano *et al.*, "A review of agent and service-oriented concepts applied to intelligent energy systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1890–1903, aug 2014.

[23] S. Karnouskos, R. Sinha, P. Leitao *et al.*, "Assessing the integration of software agents and industrial automation systems with ISO/IEC 25010," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, Jul. 2018.

[24] ——, "The applicability of ISO/IEC 25023 measures to the integration of agents and automation systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2018.

[25] L. Ribeiro, S. Karnouskos, P. Leitao *et al.*, "Performance assessment of the integration between industrial agents and low-level automation functions," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, Jul. 2018.

[26] P. Pop, M. L. Raagaard, M. Gutierrez *et al.*, "Enabling fog computing for industrial automation through time-sensitive networking (TSN)," *IEEE Commun. Standards Magazine*, vol. 2, no. 2, pp. 55–61, Jun. 2018.

[27] M. Rihan, M. Elwekeil, Y. Yang *et al.*, "Deep-VFog: When artificial intelligence meets fog computing in v2x," *IEEE Systems Journal*, pp. 1–14, 2020.

[28] K. Tange, M. D. Donno, X. Fafoutis *et al.*, "A systematic survey of industrial internet of things security: Requirements and fog computing opportunities," *IEEE Commun. Surveys & Tutorials*, pp. 1–1, 2020.