

Common Practices for Integrating Industrial Agents and Low Level Automation Functions

Paulo Leitão*, Stamatis Karnouskos[†], Luis Ribeiro[‡], Panayiotis Moutis[§], José Barbosa*, Thomas. I. Strasser[¶]

*Polytechnic Institute of Bragança, Campus Sta Apolónia, 5300-253 Bragança, Portugal, email: {pleitao, jbarbosa}@ipb.pt

[†]SAP, Karlsruhe, Germany, email: stamatis.karnouskos@sap.com

[‡]Linköping University, SE-58 183 Linköping, Sweden, email: luis.ribeiro@liu.se

[§]Carnegie Mellon University, Pittsburgh, PA, United States, email: pmoutis@andrew.cmu.edu

[¶]AIT Austrian Institute of Technology, Vienna, Austria, email: thomas.strasser@ait.ac.at

Abstract—Industrial agent technologies have been integrated in key elements coupling industrial systems and software logic, which is an important issue in the design of cyber-physical systems. Although several efforts have been tried out over the last decades to integrate software agents with physical hardware devices, and some commonalities can be observed among the existing practices, there is no uniform way overall. This work presents an empirical survey of existing practices in three application area, namely factory automation, power & energy systems and building automation. It identifies pertaining common issues and discusses how they integrate low level automation functions by utilizing industrial agents. The surveyed practices reveal high diversity, customized traditional integration focusing mostly on I/O functions, without security, and an overall approach that is mostly coupled rather than embedded.

I. INTRODUCTION

Industrial systems are being re-designed to address the continuous globalization and digitalization of economies of scale, and the growing demand for more complex but in parallel highly-customized products [1], [2]. Industrial Agents (IAs) [3] have been used to introduce intelligence and adaptation in such complex dynamic systems, constrained by some industrial requirements, namely the need to integrate low level automation functions, with lighthouse examples in several domains [1], such as factory automation, power & energy systems, and building automation.

Of special significance to IAs, are Multi-Agent Systems (MAS) which is a paradigm derived from the distributed artificial intelligence field that promote distribution, decentralization, intelligence, autonomy and adaptation, contributing to achieve flexibility, robustness, responsiveness and reconfigurability [4]. A MAS is an ecosystem of intelligent, autonomous and cooperative computational entities, known as agents, which may represent physical or logical objects in the system. The overall system function emerges from the interaction among distributed agents, each one possessing its own knowledge and skills, and able to interact with other agents when it can achieve its goals. MAS offers an alternative way to design complex large-scale systems by decentralizing the control system by distributed, autonomous and cooperative entities, differing from the conventional approaches due to its

inherent capabilities to adapt to emergence without external intervention.

MAS technology has already been integrated to several industrial applications in a Cyber-Physical Systems (CPS) context, namely smart production [3], [5]–[10], smart power grids [11]–[14]), smart logistics [6]) and smart healthcare [15], [16]). IAs may form an integral part of CPS and are well aligned with the trends in the domain [1]–[3], expand the potential application domains of MAS and at the same time add the required flexibility, robustness and responsiveness to industrial automation systems. However, as recent surveys reveal [4], [17]–[20], their acceptance depend on key factors, including technology, standardization and hardware, which pertain the investigation in this work.

Although significant efforts exist in IAs, the practices to integrate the software agent counterpart with the physical hardware device counterpart are not homogeneous. Different domains have focused on different practices, that pertain the usage of tools, integration consideration and approach. Despite of this, some general commonalities can be observed. In this work, a survey was undertaken in three of such domains, i.e., factory automation, power & energy systems, and building automation. The aim was to collect empirical data on current practices and identify commonalities, that posteriorly will support the task to potentially propose a “best practice” for that specific domain or integration of IAs and low-level automation functions at large.

The following part of the paper is organized as follows: in [section II](#) the methodology used to perform the survey related to existing interfacing practices, namely explaining the selected application domains and the templates to categorize the analyzed practices, is described. Subsequently, [section III](#) analyses the survey results for the three independent application domains, namely factory automation, power & energy systems, and building automation. In [section IV](#) a discussion is carried out with the aim to identify commonalities and generalize the results towards the establishment of “best practices” for integrating IAs and low level automation functions at large. Finally, in [section V](#) the conclusions and future work are presented.

II. METHODOLOGY

As outlined in the introduction, the focus of this study is centered on three industrial domains, i.e., factory automation, power & energy systems, and building automation, as these form the core of efforts for integrating software agents, and they are also prevalent in ongoing efforts related to key high level initiatives in many countries, such as Smart Grids, Smart Cities and Fourth Industrial Revolution (Industry 4.0). To identify the existing approaches in integrating software agents with low level automation functions, a survey was selected as a mean of collecting data.

A questionnaire was constructed to clearly focus to a single practice of interconnecting the agent and the automation device. The questionnaire attempts to capture key aspects of such interfaces, e.g., the type of devices/controllers used, the interface concept, the type of services used, the data model of the interface, technologies & protocols utilized, etc. as well as a self-assessment of benefits & weaknesses for the specific practice. Note that the interface approach can be classified as follows (and as shown in Figure 1):

- *Coupled approach*: Software agents are running remotely and accessing the physical automation device, that is running a native logic control layer that ensures (real-time) responsiveness, to get data or send commands.
- *Weak embedded approach*: Software agents are directly embed in automation devices (sensors, actuators, Programmable Logic Controllers (PLCs), and microcontrollers), but are accessing the native logic control layer that ensures (real-time) responsiveness.
- *Strong embedded approach*: Software agents are directly embed in automation devices (sensors, actuators, PLCs, and microcontrollers) and are running the logic control of the device, e.g., by accessing input and output ports.

The sample is exploratory and can be categorized as convenience sampling. The primary input stems from the participants of the IEEE P2660.1 working group [21].

Following the acquisition of the empirical data collection via questionnaires, these were firstly analyzed within their respective domain, in order to obtain domain-specific insights. Subsequently, the results were looked from a domain-agnostic viewpoint, in order to identify commonalities, such as technologies, interfaces, practices etc. Having obtained the expected insights, a critical discussion follows identifying pros and cons, potential challenges and future directions.

The work carried out has several limitations. It should be pointed out that this survey does not aim at an exhaustive capturing of all practices in each domain, but some common ones. In addition, the focus is limited on the three aforementioned domains, while several others are considered as future work e.g., logistics. Another limitation of the approach is that since the sampling focuses on the IEEE P2660.1 group, it is clear that the results will reflect overwhelmingly the views of research institutes, universities and research departments within companies. Such limitations are seen as acceptable, as

the aim is to gain some initial insights and the results may be the starting point for future more in-depth studies.

III. SURVEY RESULTS

Three domains were indicatively surveyed as analyzed in this section. An overview of the results is depicted in Table I.

A. Factory Automation

Eight practices (F1–F8) as shown in Table I have been analyzed in factory automation. Their Technology Readiness Level (TRL) [22] is positioned between TRL3 and TRL9, while most are of TRL4. The levels of solution robustness and technical availability are generally low. Out of the eight documented practices, 6 approaches match the coupled template (F1–F6), while two match the weak embedded template (F6–F7) and one the strongly embedded template (F8).

The analysis shows that regarding the practices that match the coupled template mostly target the interaction with conventional PLCs and, not surprisingly the functions typically used align in this direction and include: handling programs, digital I/O read/write, analog I/O read/write and the notification of events. Several custom data types are considered. Java is the prevalent technology on the agent side and, as mentioned, PLCs supporting IEC 61131-3 are frequently used without any vendor preference. The agent to PLC interaction mechanism makes use of conventional communication protocols. The practices matching the weak embedded template follow a similar line but here on practice reported the use of Raspberry Pi controllers. Only one strong embedded practice was reported and in this instance a custom software stack was developed for native applications supported by the JAMAICA VM over Linux. Overall the generic interfacing aspects do not seem to be mature.

B. Power & Energy Systems

Four practices (P1–P4) were analyzed related to power & energy systems (see Table I). All of the practices can realize a coupled interconnection of the agent with the automation device, while P3 can also facilitate the embedded paradigm. The technological maturity varies considerably among them: one is found at TRL 6 and the rest between TRL 2 and 4.

The analysis indicates that the coupled interface practices are based on libraries of software that are incorporated as directories of the host program defining the agent entity. On the automation device side, IEC 61499 and IEC 61850 are the basis for practices P2 (potentially) and 3, IEC 61499 and IEC 61131-3 are used in practice P4, while Modbus is preferred in practice P1. Monitoring and controlling digital I/O variables is common ground in all four practices. Practice P2 can be described as the most interactive of the ones gathered in this domain, since it can go as far as handle the execution of programs on the automation device side, if the agent requires so. Handling of event notifications in practices P3 and P4 may be attributed to the well-defined Application Programming Interface (API) that is available in these interface examples.

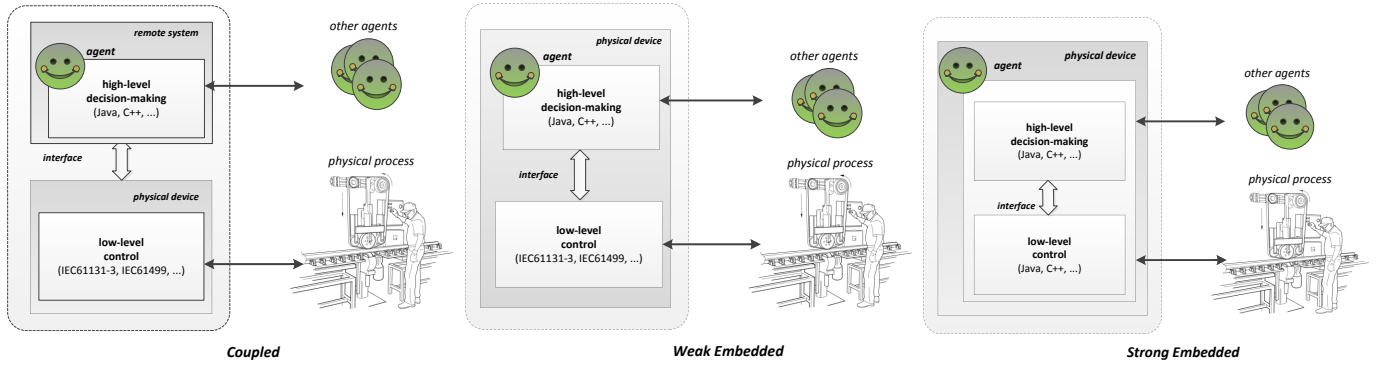


Figure 1. Interfacing practices: Coupled, Weak Embedded and Strong Embedded.

All of the practices, regardless of coupled or embedded operation with the agent, are programmed in or as high-level languages making them easier to use. It is interesting to note that practice P2 interacts with the device through a USB communication channel and practice P4 on a RS-232 protocol. The surveyed experts' opinion on these practices pointed out the opportunity for promoting them as readily applicable agent-automation interfaces, although the security of their implementation is a matter of concern.

One fifth interface practice (not analyzed separately Table I) is the OpenMUC software framework [47]. OpenMUC, when enabled over the Modbus/TCP communication protocol, is built over JaMod (practice P1).

C. Building Automation

A total of five practices (B1–B5) as shown in Table I have been analyzed in the building automation domain. Their TRL is positioned between TRL3 and TRL9 with a prevalence of TRL4 and the levels of solution robustness and technical availability are generally low or moderate. Out of the five documented practices, two approaches match the coupled template (B1–B2), with one being considered as loose-coupled since it uses a broker approach, while B3–B5 match the weak embedded template.

The analysis shows that regarding the practices that match the coupled template target the interaction with monitoring of home environmental parameters and the control of devices, namely those considering the KNX protocol. Not surprisingly the functions typically include: digital I/O read/write, analog I/O read/write and the notification and subscription of events. Several custom data types are considered. Java is the prevalent technology on the agent side and, diverse device types are frequently used. The agent-to-device interaction mechanism make use of conventional communication protocols, particularly the ones based on TCP/IP. Both physical and wireless communication environments were utilized while the last is also considering the trending use of the publish/subscribe architectural approach. The practices matching the weak embedded template follow a similar line, i.e., in the diversity of device types. Here, the practices reported make use of newly devices, particularly the Raspberry Pi controllers, where it is

possible the direct deployment of the agent compiled execution code. Overall the generic interfacing aspects do not seem to be mature, except the one offered by a commercial company.

IV. DISCUSSION

A first observation on the surveyed interface practices is that Java is predominantly utilized as the agent-side technology, followed by C++. This emanates a requirement for interfaces that can be incorporated as high-level programming appendices – most probably as libraries. In turn, such a requirement raises the question of whether a “best practice” proposal should also suggest how such a collection of appendices should be organized, both in terms of content and architecture.

Most analyzed interfaces follow the coupled approach, i.e., remote access to the physical hardware device. Some interfaces prefer the weak embedded template, while the strong embedded interface is barely considered. The aforementioned statistics imply that, usually, the interface of industrial agents in the context of CPS is not used for (hard) real-time control applications. This may be attributed to the general characteristics of services, control and operations, which the overlying agent system implements. If the events and actions that the agents respond to are in the time-frame of seconds to minutes (or more), an embedding interface practice that can handle fast implementation (compared to coupled) is, practically, obsolete. Although one could object that interface practices that enable strong embedding would offer maximum flexibility, the argument that customization of the interface to any specific automation device is required, counters this approach for multiple reasons.

From the perspective of information exchanged over the interface, it seems that all practices, at the absolute minimum, can handle read/write operations on the digital I/O interaction of the automation device with the controlled assets. All other device services vary from one interface to another, although some characteristics may attest to underlying trends (e.g., an API-enabled interface tends to be capable of some handling of events monitored through the device). Since, the digital I/O interaction of the automation device can be considered as a basic operation, all interface practices of such devices with a controlling agent should agree on a standard protocol [48]

Table I
SUMMARY OF THE ANALYZED PRACTICES IN FACTORY AUTOMATION (F1-F8), POWER & ENERGY (P1-P4), AND BUILDING (B1-B5) DOMAINS.

ID	Application Example	Typical Functions	Agent Technologies	Devices Used	Interface Technologies	Info	Type
F1	Discrete Process Control of Robotic Cells	Handling Programs - Get Status; Digital I/O read/write; Subscription and notification of events.	Java	PLC (61131-3) (Modicon M340, Omron CPM1)	OPC with unspecified or custom data types.	[23], [24]	coupled
F2	no specific case – collection of proprietary practices	Several Unspecified	Java, JSON	Several Unspecified	OPC-UA with unspecified or custom data types.	[25]–[30]	coupled
F3	Unspecified Device Control	Handling Programs - Start, Stop, Upload, Select, Get Status; Digital I/O read; Analog I/O read.	C/C++	PLC (61131-3); Raspberry Pi	EtherCAT with several data types including: unspecified, CAEX (IEC 62424), AutomationML (IEC 62714), binary, program data, product data, process data, maintenance data.	[31]	coupled
F4	Vehicle Motion Control	Handling Programs - Start, Stop, Select, GetStatus; Digital I/O read/write; Analog I/O read/write; Notification of events.	C/C++, MATLAB / SIMULINK	PLC, Single-board computer	Unspecified with several data types including: binary, program data, process data.	[32]	coupled
F5	Unspecified Device Control	Several Unspecified	Java	PAC (61131-3)(Wago)	MODBUS	[33]	coupled
F6	Unspecified PLC Control	Handling Programs - Start, Other; Digital I/O read/write, Analog I/O read/write; Notification of events.	Java	PLC (61131-3) (ELREST Combo CM211), Other	TCP/IP, Other with several data types including: unspecified, binary, program data, product data, process data.	[34], [35]	coupled, weak embedded
F7	Discrete Process Control of Robotic Cells	Handling Programs - Start, Stop; Digital I/O read/write; Analog I/O read/write; Notification of events; Subscription of events; Other.	Java	Raspberry Pi (PI4J API)	Custom Solution	[23], [24]	weak embedded
F8	Unspecified Device Control	Several Unspecified	Java, C/C++	Unspecified	JAMAICA VM, Runtime, Linux	[36]	strong embedded
P1	JaMod	Digital I/O read/write; Analog I/O read.	Java	Advantech devices	TCP/IP Modbus.	[37]	coupled
P2	mbed library	Handling Programs - Start, Stop; Digital I/O read/write; Analog I/O read/write; Handling events Notification/Subscription.	C++ over LabVIEW	mbed micro-controller	USB.	[38], [39]	coupled
P3	nxtControl & nxtSTUDIO	Digital I/O read/write; Analog I/O read/write; Notification of events.	IEC61499 & IEC61850	PLC	UDP or embedded.	[40]	coupled & strong embedded
P4	Active power distribution control	Digital I/O read/write; Analog I/O read/write; Notification of events.	Java	Inverter-battery charger and intelligent load controller (PLC-type; IEC 61131-3 or IEC 61499)	OPC UA-based; access to inverter controller via XML-RPC (over RS-232) and OPC UA/DA gateway; access to inverter controller via OPC UA/PLC gateway.	[41], [42]	coupled
B1	KNX system control	Digital I/O read/write; Analog I/O read/write; Notification of events.	Java	KNX devices	KNXnet/IP with unspecified data model, including binary data.	N/A	coupled
B2	Room temperature monitoring and control	Notification and subscription of events.	Java	Diverse atmospheric sensors	Message Queuing Telemetry Transport (MQTT) with unspecified data model, including binary data.	[43]	coupled
B3	Room light brightness control	Handling Programs - Start, Stop, Upload, Select, GetStatus; Digital I/O read/write; Analog I/O read/write; Notification of events and Management commands in XML format.	IEC61499, C++, JSON	Softlink Ideabox 3 (IEC61499)	Unspecified but including binary, program and process data.	N/A	weak embedded
B4	Water meter PLC Control	Handling Programs - Start, Stop; Digital I/O read/write.	Unspecified with JSON format.	Unspecified	TCP/IP Modbus with MultiSpeak standardized data model	N/A	weak embedded
B5	Home Automation based on a Wireless Sensor Network	Handling Programs - Start, Stop, GetStatus; Digital I/O read/write; Analog I/O read/write; Notification of events.	C/C# (IEC61499)	Raspberry Pi, TelosB, CC2530	Unspecified but including binary data.	[44]–[46]	weak embedded

and/or a common ontology, regardless of the programming language or other parameters.

A pertinent question that arises from this analysis is that more elaborate services are not used, e.g., handling programs, as well as the publish/subscribe mechanisms. This probably can be justified by typical functions associated to software agents when interacting with low-level automation functions, but is also justified by the immature technologies to support the interfacing. At the moment, the emergence of Industrial Internet of Things (IIoT) technologies offering a plethora of new capabilities, e.g., in terms of event-driven mechanisms, will permit to extend the use of more elaborate functions. Furthermore, it's expected that, for particular tasks, the agent's interfaces practices will be adjusted to future evolution in the IoT architectural trends, following possibly a more loosely-coupled approach where the agent-device interface is done through communication brokers.

The reported interfaces use usually a kind of proprietary protocols, but some others prefer the use of more standardized ones, such as OPC UA or IEC 61850. This last option is preferred since it facilitates the industrial adoption. In fact, ISO 9506 [49], known as MMS (Manufacturing Message Specification), was a predecessor in this field by defining the application layer of the ancient MAP (Manufacturing Automation Protocol), and consequently providing a platform capable to interconnect various industrial computerized devices supplied by different suppliers. The guidelines defined by ISO 9506 can be used as source of inspiration in this work.

A major concern noticed by the surveyed experts is the security of the interface practices collected, which may be further amplified as the controlled devices may rely in critical infrastructures. Ensuring that both the agent decisions passed through the practice to the automation device, as also the feedback from the device to the agent should be ascribed to some security best practices and risk analysis procedures.

Apart from the general lack of common APIs and information exchanged [48], another issue raised is that of compatibility testing, especially when this pertains cross-language interactions as well as the interactions at agent or device level. This is mostly a result of lack of common APIs and best practices. However, the adoption of a testing approach e.g., based on ISO-29119 [50] set of standards, may enhance interoperability both at agent-to-physical device level via their interface, as well as between the agent-to-agent interactions. Such software tests, could also be expanded to concrete use-cases and industries, depending on their requirements. The overall result would be not only to have a "best practice" on how to couple industrial agents with low level automation functions, but also detailed testing suites and processes that guarantee the applicability of these practices and the interoperability and quality on software life-cycle. Such testing approach ought to complement the technology specific ones that might exist for each practice. It has to be pointed out though, that this might be a challenging undertaking, as even ISO-29119 faces critique on aspects pertaining context driven testing.

V. CONCLUSIONS

This paper describes a survey undertaken to analyze the current practices on interfacing software agents and low-level automation functions, in a CPS context. The collected empirical data for three domains, i.e., factory automation, power & energy systems, and building automation, allowed to identify commonalities that posteriorly will support the proposition of a "best practice".

The main observations extracted from the analysis of the reported interfaces are the overwhelming use of Java and C++ as programming languages to implement the software agents, the preferable use of the couple approach, the use of proprietary protocols for the interface, the focus on I/O functions and the missing security issues in the reported practices. However, probably the main conclusion extracted from this survey is that the final result would be not only to have a "best practice" on how to couple industrial agents with low level automation functions, but instead a set of "best practices" designed according to the domain and the application characteristics. This requires testing methods to support a recipe to select the best interfacing practice according to several input parameters that defines the application scenario.

Future work will be devoted to enlarge the sampling, particularly capturing other views from research institutes, universities and research departments within companies, and to derive the "best practices" to implement the interfacing between software agents and low level automation functions at large and for each specific domain.

REFERENCES

- [1] 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, May 2016.
- [2] S. Karnouskos, A. W. Colombo, and T. Bangemann, "Trends and challenges for cloud-based industrial cyber-physical systems," in *Industrial Cloud-based Cyber-Physical Systems: The IMC-AESOP Approach*. Springer, May 2014.
- [3] P. Leitão and S. Karnouskos, Eds., *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, Mar. 2015.
- [4] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979–991, Oct. 2009.
- [5] A. W. Colombo, T. Bangemann, S. Karnouskos, J. Delsing, P. Stluka, R. Harrison, F. Jammes, and J. L. Martínez Lastra, Eds., *Industrial Cloud-based Cyber-Physical Systems: The IMC-AESOP Approach*. Springer, 2014.
- [6] P. Leitão, V. Mařík, and P. Vrba, "Past, present, and future of industrial agent applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2360–2372, Nov. 2013.
- [7] L. Monostori, J. Váncza, and S. Kumara, "Agent-based systems for manufacturing," *CIRP Annals - Manufacturing Technology*, vol. 55, no. 2, pp. 697–720, Jan. 2006.
- [8] V. Mařík and D. McFarlane, "Industrial adoption of agent-based technologies," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 27–35, Jan. 2005.
- [9] V. Mařík and J. Lažanský, "Industrial applications of agent technologies," *Control Engineering Practice*, vol. 15, no. 11, pp. 1364–1380, Nov. 2007.
- [10] A. W. Colombo, S. Karnouskos, J. M. Mendes, and P. Leitão, "Industrial agents in the era of service-oriented architectures and cloud-based industrial infrastructures," in *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, 2015, pp. 67–87.

- [11] T. Strasser, F. Andren, J. Kathan, C. Cecati, C. Buccella, P. Siano, P. Leitão, G. Zhabelova, V. Vyatkin, P. Vrba, and V. Mařík, "A review of architectures and concepts for intelligence in future electric energy systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2424–2438, Apr. 2015.
- [12] P. Vrba, V. Mařík, P. Siano, P. Leitão, G. Zhabelova, V. Vyatkin, and T. Strasser, "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.
- [13] S. Karnouskos, A. Weidlich, J. Ringelstein, A. Dimeas, K. Kok, C. Warmer, P. Selzam, S. Drenkard, N. Hatzigiorgiouris, and V. Lioliou, "Monitoring and control for energy efficiency in the smart house," in *1st International ICST Conference on E-Energy, 14-15 October 2010 Athens Greece*, 2011, pp. 197–207.
- [14] S. Karnouskos, "Cyber-Physical Systems in the SmartGrid," in *9th IEEE International Conference on Industrial Informatics (INDIN'11)*, Jul. 2011, pp. 20–23.
- [15] M. Furmankiewicz, A. Softysik-Piorunkiewicz, and P. Ziuziański, "Artificial intelligence and multi-agent software for e-health knowledge management system," *Informatyka Ekonomiczna*, vol. 2, 2014.
- [16] F. Bergenti and A. Poggi, "Multi-agent systems for e-health: Recent projects and initiatives," in *10th International Workshop on Objects and Agents*, 2009.
- [17] P. Leitão and S. Karnouskos, "A Survey on Factors that Impact Industrial Agent Acceptance," in *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, Mar. 2015, pp. 401–429.
- [18] S. Karnouskos and P. Leitão, "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.
- [19] J. P. Müller and K. Fischer, "Application Impact of Multi-agent Systems and Technologies: A Survey," in *Agent-Oriented Software Engineering*. Springer, 2014, pp. 27–53.
- [20] M. Metzger and G. Polakow, "A survey on applications of agent technology in industrial process control," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 570–581, Nov. 2011.
- [21] IEEE Project P2660.1 – Recommended Practices on Industrial Agents: Integration of Software Agents and Low Level Automation Functions. [Online]. Available: <https://standards.ieee.org/develop/project/2660.1.html>
- [22] European Commission Technology Readiness Levels (TRL). HORIZON 2020, General Annexes, Extract from Part 19 – Commission Decision C(2014)4995. [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf
- [23] P. Leitão and J. Barbosa, "Building a robotic cyber-physical production component," in *Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 2016, pp. 295–305.
- [24] J. Dias, J. Barbosa, and P. Leitão, "Deployment of industrial agents in heterogeneous automation environments," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, Jul. 2015.
- [25] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, Elsevier, 2014.
- [26] F. Jammes, S. Karnouskos, B. Bony, P. Nappey, A. W. Colombo, J. Delsing, J. Eliasson, R. Kyusakov, P. Stluka, M. Tilly, and T. Bangemann, "Promising Technologies for SOA-Based Industrial Automation Systems," in *Industrial Cloud-based Cyber-Physical Systems: The IMC-AESOP Approach*. Springer, 2014, pp. 89–109.
- [27] T. Bangemann, S. Karnouskos, R. Camp, O. Carlsson, M. Riedl, S. McLeod, R. Harrison, A. W. Colombo, and P. Stluka, "State of the Art in Industrial Automation," in *Industrial Cloud-based Cyber-Physical Systems: The IMC-AESOP Approach*. Springer, 2014, pp. 23–47.
- [28] S. Karnouskos and V. Somlev, "Performance assessment of integration in the cloud of things via web services," in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 1988–1993.
- [29] S. Karnouskos, V. Villaseñor-Herrera, M. Haroon, M. Handte, and P. J. Marrón, "Requirement considerations for ubiquitous integration of cooperating objects," in *2011 4th IFIP International Conference on New Technologies, Mobility and Security*, Feb. 2011, pp. 1–5.
- [30] S. Karnouskos, "Realising next-generation web service-driven industrial systems," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 1-4, pp. 409–419, Sep. 2011.
- [31] myjoghurt demonstrator. Technical University of Munich. [Online]. Available: <http://i40d.ais.mw.tum.de/>
- [32] C. Chin and S. Lum, "Rapid modeling and control systems prototyping of a marine robotic vehicle with model uncertainties using xPC target system," *Ocean Engineering*, vol. 38, no. 17-18, pp. 2128–2141, Dec. 2011.
- [33] T. Berger, D. Deneux, T. Bonte, E. Cocquebert, and D. Trentesaux, "Arezzo-flexible manufacturing system: A generic flexible manufacturing system shop floor emulator approach for high-level control virtual commissioning," *Concurrent Engineering*, vol. 23, no. 4, pp. 333–342, Jul. 2015.
- [34] L. Ribeiro and P. Linder, "Hardware abstraction layer for JAVA-based agents," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2016.
- [35] L. Ribeiro and J. Barata, "Deployment of multiagent mechatronic systems," in *Lecture Notes in Computer Science*. Springer, 2013, pp. 71–82.
- [36] S. Theiss, "Echtzeitfähige softwareagenten zur realisierung cyber-physischer produktionssysteme," Ph.D. dissertation, Technische Universität Dresden, 2015.
- [37] Java Modbus Library (jamod). [Online]. Available: <http://jamod.sourceforge.net/>
- [38] C. L. Moreira, D. Rua, E. Karfopoulos, E. Zountouridou *et al.*, "Extend concepts of MG by identifying several EV smart control approaches to be embedded in the Smart Grid concept to manage EV individually or in clusters," *Mobile Energy Resources in Grids of Electricity (MERGE) project*, Tech. Rep., 2010, deliverable D1.2. [Online]. Available: <https://goo.gl/4G9SGY>
- [39] S. Skarvelis-Kazakos, P. Papadopoulos, I. G. Unda, T. Gorman, A. Belaidi, and S. Zigan, "Multiple energy carrier optimisation with intelligent agents," *Applied Energy*, vol. 167, pp. 323–335, Apr. 2016.
- [40] G. Zhabelova, V. Vyatkin, and V. N. Dubinin, "Toward industrially usable agent technology for smart grid automation," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2629–2641, Apr. 2015.
- [41] M. Merdan, A. Prostejovsky, I. Hegny, W. Lepuschitz, F. Andrén, and T. Strasser, "Power distribution control using multi-agent systems," in *Recent Advances in Robotics and Automation*. Springer, 2013, pp. 323–333.
- [42] A. Prostejovsky, M. Merdan, G. Schitter, and A. Dimeas, "Demonstration of a multi-agent-based control system for active electric power distribution grids," in *2013 IEEE International Workshop on Intelligent Energy Systems (IWIES)*, Nov. 2013.
- [43] J. B. Javier de las Morenas and P. Leitao, "Low Cost Integration of IoT Technologies for Building Automation," 2017, technical Report, Unpublished.
- [44] R. Abrishambaf, J. Cabral, J. Monteiro, and M. Bal, "An energy aware design flow of distributed industrial wireless sensor and actuator networks," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, Mar. 2015.
- [45] R. Abrishambaf, M. Hashemipour, and M. Bal, "Integration of wireless sensor networks into the distributed intelligent manufacturing within the framework of IEC 61499 function blocks," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2011, pp. 3458–3462.
- [46] —, "Structural modeling of industrial wireless sensor and actuator networks for reconfigurable mechatronic systems," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5-8, pp. 793–811, Mar. 2012.
- [47] OpenMUC framework. [Online]. Available: <http://www.openmuc.org/openmuc/user-guide/>
- [48] I. Seixas and P. Leitão, "Standards compliance in industrial agents applications," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Nov. 2013.
- [49] *Industrial Automation Systems – Manufacturing Message Specification, Part 1 – Service Definition*, ISO Std. ISO/IEC9506-1, 2003. [Online]. Available: <http://www.iso.org/standard/37079.html>
- [50] *ISO-29119 Software Testing*, ISO/IEC/IEEE Std. [Online]. Available: <http://www.softwaretestingstandard.org>