Key Directions for Industrial Agent Based Cyber-Physical Production Systems

Stamatis Karnouskos*, Luis Ribeiro†, Paulo Leitão‡, Arndt Lüder§, and Birgit Vogel-Heuser¶

*SAP, Germany, email: stamatis.karnouskos@sap.com
†Linköping University, Sweden, email: luis.ribeiro@liu.se
‡Research Centre in Digitalization and Intelligent Robotics (CeDRI),
Instituto Politécnico de Bragança, Portugal, email: pleitao@ipb.pt
§Otto von Guericke Universität Magdeburg, Germany, email: arndt.lueder@ovgu.de
¶Automation and Information Systems, Technical University of Munich, Germany, email: vogel-heuser@tum.de

***Abstract*—Rapid technology advances and increasing digitalization bring new opportunities but also put new challenges to modern production systems. Industrial agent concepts and technologies can be used to address the needs of modern Cyber-Physical Production Systems (CPPS). To do so, their interplay, especially in industrial settings, needs to be better understood and assessed. Key directions for investigations that are relevant for industrial agent based CPPS are analyzed in this work, mainly the design patterns, interfaces between agents and CPPS, metrics to evaluate the quality of agent based CPPS, and distributed intelligence implemented in or by agents. Although efforts exist both in agent and CPPS domains, their amalgamation still has several aspects that are under-investigated, especially when considering large-scale systems of CPPS, and the proposed four directions discussed in this work, seem promising to address several of the key underlying challenges.**

I. Introduction

Modern production systems are facing new challenges as they have to cope with an increasing global competition and rapid technological developments. These trends result in a growing complexity for both products and production systems, something that also reinforces the pivotal roles of engineering, deployment, and operation of production systems. To address the challenges brought from these trends, new production system architectures with advanced monitoring & control are envisioned [1]–[6] that capitalize on the capabilities of the new technologies. Initiatives such as RAMI/Industrie 4.0 [7] and Industrial IoT [8], to name two of the most prominent, have in their core similar ideas i.e. exploiting the latest IT technologies, concepts, and infrastructure, to realize distribution of control decision-making.

To reduce the complexity of engineering, implementation, and use of these complex network control systems, the developed architectures are based around the idea of Cyber-Physical Production Systems (CPPS) [2]. CPPS are envisioned as production system components with information processing and communication/interaction capabilities able to execute physical processes within a production system in cooperation with other entities. Each CPPS component has to take the necessary control decisions related to its underlying production system physical aspects and to communicate control decisions, system states and behavior patterns. To implement the necessary information processing and exchange required to enable CPPS components to take their control decisions, different architecture patterns and implementation technologies have been developed and applied. They range from service-oriented architectures exploiting technologies like web services to agent-based architectures exploiting FIPA-compliant (Foundation for Intelligent Physical Agents [9]) solutions. However, they come also with their own set of challenges.

One example of an architecture pattern applicable to implement CPPS is multi-agent systems (MAS) [10] and more specifically Industrial Agents [3], [11] that addresses specific industry requirements in productive systems. MAS expose system characteristics like autonomy, cooperation, intelligence, reactivity, and proactivity, allowing to distribute intelligence among a network of control nodes, and consequently being effectively tailored to distributed control systems, namely implementing CPPS solutions [3]. While MAS for control can be considered as a mature architecture pattern, their application in industry is still limited [3], [12].

With the increased adoption of CPPS, also the use of agent-based control can be reconsidered for their application in industrial domains. To support this objective, this work discusses four main challenges to the use of agent-based systems within CPPS and their control, namely focusing the patterns, interfaces, metrics and distributed intelligence. Therefore, the paper is structured as follows: Section II overviews the relevant requirements for the development of CPPS and Section III describes the key directions to address these requirements by using intelligent software agents. Section IV discusses these key directions, especially aligning with the RAMI reference architecture and finally, Section V rounds up the paper with the conclusions and points out some future work.

II. Requirements for Industrial Agent based CPPS

The definition of relevant requirements for the development of CPPS is a complex exercise and contributions are spread across roughly twenty years in literature. Therein both functional and non-functional requirements have been formulated with different degrees of formality, considering different views on and abstractions of CPPS predecessors. These requirements have originated from both top-down and bottom-up design processes, with the first being predominant in traditional automation system design, while the second has been normally considered in the design of agent-based and automation solutions.

One pertinent challenge is that, despite the potential shown for supporting highly adaptable systems, agent-based approaches have elusively been applied outside prototype production systems. This had an impact on understanding the main requirements that are imposed on them. This is in contradiction to other existing automation solutions in operation which are constantly exposed to requirements emerging from practice. Furthermore, the operation of the conventional automation system has led to a generally accepted and stratified automation pyramids isolating different requirements and concerns purely from a logical point of view. However, the concepts surrounding CPPS imply that a functional entity in a production system has a harmonized logical and physical existence (cyber-physical view). In this case, is it also very likely that such entity occupies, as a unit, several layers within the traditional automation pyramid.

However, there is currently a general lack (or incipient set) of models that could accurately describe such systems. In the traditional automation pyramid, software and supporting computing platforms occupy one layer as a function of the time frame where they need to operate. Emerging cyber-physical solutions encourage a new way of thinking where structural and logical changes are part of the system's operations and can be enabled by mobile, highly reconfigurable, or pluggable modular components. They have generally adhered to the agent, or holonic principles of organization [13] materialized in several well-known reference architectures.

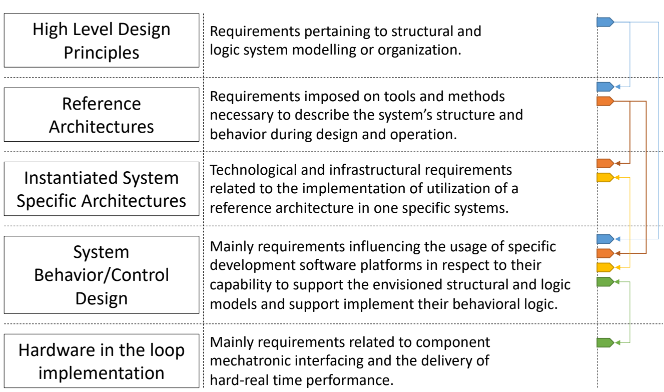

Figure 1. Design Stages

Requirement specification for the agent or holonic-based systems, as a systematic exercise to develop CPPS, has however been incomplete and failed to connect these reference architectures to industry-grade software, able to operate over industrial production equipment [3], [4], [14]–[16]. This challenge is more evident in the design process as it cuts across the several conceptual stages shown in Figure 1, where researchers have generated requirements belonging to the different stages but have followed different paths concerning implementation and demonstration (denoted by the colored lines in Figure 1). The lack of an all-encompassing design procedure, consistently connecting requirements generated at the different design stages, has affected decisively the acceptance potential of CPS-based solutions for industrial systems. Without it, the benefits generated by the CPS abstractions envisioned when discussing abstract high-level design principles are never really present within the final connected implementation.

The literature survey shows that it is possible to isolate consistently a set of common high-level design principles and proceed with their formal formulation [14]. These common requirements adhere to others already existing for systems in operation namely: Predictability, Usability, Diagnosability, Safety, and Security. However, especially in the past 20 years, Adaptability, Integrability, and Convertibility have emerged as specific top-level requirements that pertain to the new ways of modeling, developing, implementing and operating production systems now commonly known as CPPS [4], [14]. These aspects are important for the adoption of agents in industry [3], [12], and are also seen as critical in addressing key issues of CPS systems at large [5], [6].

While Adaptability captures the behavioral autonomy of the system, it cannot operate without several layers of Integrability and Convertibility. The former ensures the logical connectivity between components within a system and allows autonomous response to propagate logically while the latter ensures that this propagating logical response is translated into physical actuation. Collectively Autonomy, Integrability, and Convertibility can be seen as the three functional pillars of CPPS design and operation. These can be broken down into additional requirements along the system design process, which makes them relatively easy to specify as abstract constructs. However, their implementation and integration have very concrete challenges which currently still encounter important technological and conceptual constraints. In this context, it is valuable to understand what are the main design patterns, system integration interfaces and abstractions, application of AI and metrics that should be considered to design, development and operate CPPS.

III. Industrial Agent Based CPPS Directions

As discussed in the previous section, four main key directions are identified and discoursed for industrial agent based CPPS, namely: Design Patterns for Agents and CPPS, Interfaces for Agents and CPPS, Metrics for Solution Assessment, and Distributed Intelligence & Machine Learning.

A. Design Patterns for Agents and CPPS

Design patterns occur at all of the stages shown in Fig. 1, and when aligned, they create the much-needed cohesion to identify and consider specific solutions to different automation problems. However, given the challenges discussed in Section II, design patterns for CPPS are not easy to characterize. One first step in such characterization based on [3], [15], [17], and the analysis of more than 20 other occurring patterns is summarized in Table I (based on [18]).

Table I is for the different stakeholders to evaluate the pattern adequacy in their application case. For example, the "Application purpose & objectives" reflect potential requirements such as Adaptability, Flexibility, Plug-ability, Reconfigurability. Other criteria reflect additional requirements such

Preprint version of doi:10.1109/icphys.2019.8780360

Table I
CLASSIFICATION CRITERIA FOR FIELD-LEVEL MAS OR AGENT PATTERN (ADAPTED FROM [18])

Criteria	Description of Criteria	Examples
Purpose & objectives	What to be reached by MAS or the agent itself?	Order handling, Fault tolerance, CPPS adaptation
Pattern description	Logical MAS structure, included agents (for MAS)	Product, Process, Resource, Broker, Communication agent, etc.
Target Environment	Context of MAS or agent	Domain, Control hardware
Realization	Technical implementation	Software, Protocol
Knowledge modeling & processing	What type of model? What type of processing?	Meta model, Ontology
Autonomy	Degree of autonomy	Full, half- autonomy
Real time capabilities	Capabilities of MAS or agents actions	Hard-, Soft-realtime, no
Dependability	Availability requirements, trust	None or degree of reliability or maintainability
Knowledge acquisition	Methods and techniques to acquire knowledge, learn	Machine learning methods, e.g., neural networks
Other	Additional domain-specific requirements	Data editable in run-time

as level of system autonomy, performance, dependability, and automated learning opportunities. Patterns have implications in system structure and function. Common structural patterns include the usage of the following agents/holons [3], [18], [19]: *Resource* (to abstract and interface heterogeneous production equipment with the agent platform), *Product* (to abstract and control the production of individual items), *Order* (to represent different production orders), *Coalition/Broker* (to mediate complex interactions for example including negotiation and resource allocation).

The listed architectural constructs are reflected into implementation in a variety of ways. Hard real-time execution capabilities have been among the determinant factors. As such, "Resources" are often developed, at field level within the automatic control community [20], as native implementations that ensure deterministic performance and support the processing and execution of higher order commands. Alternatively, they can be deployed at Manufacturing Execution level with a soft real-time capable implementation connected to a hard real-time field-level gateway, which has been generally the approach of the Holonic community [3]. Both approaches have seldom been integrated despite their high complementarity.

The behavioral patterns and their implementations fulfill different requirements. Agent-based platforms offer the descriptive models and languages required for adaptive and flexible response under changes. However, these languages and models cannot handle real-time capable fault compensation or optimization within the field device layer under stringent timing requirements, and this is precisely where embedded native agents excel. When aligned, both layers can provide the correct trade-off between increased reliability through real-time support and flexible adaption on higher layers. There are however no acceptable models for systematically considering such integration and several ad hoc practices have emerged over the years [4], [21], [22]. In this context, while structural patterns have been more or less consolidating in the literature over the years, the next challenge seems to be the consolidation of the behavioral patterns, without which there is no solid base for further developments. Design patterns need to be investigated also for large-scale systems of CPPS [23], and how their interactions and collaboration can lead to next-generation infrastructure and emerging behaviors that satisfy the industrial requirements.

B. Interfaces for Agents and CPPS

Industrial agent-based solutions face the need to be compliant to strong requirements, namely specific hardware integration, reliability, fault-tolerance, scalability, industry standard compliance, and resilience [12]. Among them, interoperability has emerged as a key requirement in the design of interfaces upon which agent-based systems rely for interaction with the other systems in the factory. However, realizing it imposes several challenges such as transparency, re-usability, scalability and time behavior characteristics.

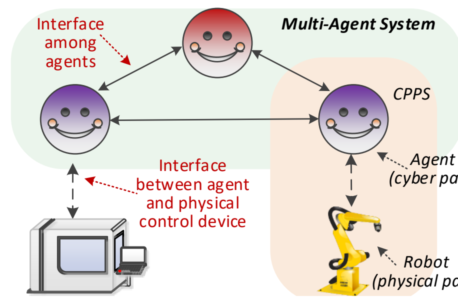

Figure 2. Interfaces in Agent-based CPPS systems

In an agent-based CPPS system, two types of interfaces can be identified (see Figure 2): i) interaction between agents and ii) interaction between the agent (cyber part) and the hardware automation control devices (physical part). For the first type, FIPA [9] has established guidelines to regulate the development of agent-based systems, and its a collection of standards that are grouped in different categories, i.e., applications, abstract architecture, agent communication, agent management, and agent message transport. The FIPA Agent Communication specifications provide the means to regulate the interaction among agents, namely focusing the Agent Communication Language (ACL) messages, interaction protocols, speech act theory-based communicative acts, and content language representations. The FIPA Interaction Protocols specifications assume a crucial relevance by defining the pre-agreed protocols for ACL messages exchanged between agents. Overall the FIPA Application specifications refer to ontology and service descriptions specifications for a particular domain. However, in the CPPS context, FIPA specifications do not sufficiently address important industry requirements such as protocols that better fit the behavior of industrial systems, event notification at the low control level, integration with legacy systems and a combination of services and agent-based systems [24]. Additionally, new and more effective collaborative models are needed for agent-based CPPS, and existing work on collaborative networks [25] can bring significant insights.

The second type of interface is related to the interconnection of the agent and the physical automation control device,

Preprint version of doi:10.1109/icphys.2019.8780360

following the HLC and LLC interaction patterns [26]. There is a need to have standardized practices that simplify and make transparent the integration process of cyber and physical counterparts. This constitutes ongoing work in IEEE P2660.1 working group, which aims to define recommended best practices [27] for using industrial agents. Several challenges are identified, namely the definition of generic templates for the interface practices, the identification of characteristics and criteria that should be considered for the assessment and comparison of the practices. In addition coherent and widely-accepted performance measurement benchmarks that will allow to compare and select the recommended practices for each specific application case are needed. However, different interface practices can be used according to the particularities of the domain application [3], and therefore it is challenging to be able to compare and recommend best practices [27], even if existing standards from the software domain and their measures can be partly adopted [28].

C. Metrics for Solution Assessment

Metrics to evaluate CPPS systems have been proposed for specific CPPS aspects e.g. an evaluation model focusing on the cyber-physical capabilities of CPS [29], or specific implemented use cases [30]. However, the question that remains is to what degree such metrics are appropriate and accurate for agent-based CPPS. As metrics are needed to be able to assess agent-based practices and compare them, their selection and applicability to agent-based CPPS [27], [28] is needed.

As an example, since agents empower flexibility and adaptivity as key characteristics of CPPS, there is a need to be able to measure them. Flexibility focuses on the capability of a production system to be adjusted to changing requirements prior to its first use and adaptability after first use [31]. In some CPPS literature [32], the term "flexibility" is used only on the sub-product level and in the segment production level. Specifically, it encompasses the ability of an entire production and logistics area to switch the production with reasonably little time and effort to new (but similar) families of components by changing manufacturing processes, material flows and logistical functions. The enhancement of flexibilities could be at the machine, material handling, process, product, routing, volume, expansion, control program, and production flexibility. However, flexibility can also be defined [33] as a non-functional requirement of a CPS and focused their differentiation on machine, process, routing, and operational flexibility. In this context, a metric to evaluate the flexibility of handling automata is proposed [34], while others have proposed [31] an analysis of the effect of changes in either product or technical process or plant (resource), Adaptivity metrics have also been imported from a software viewpoint [35], [36], focusing on the effort needed to perform adaptations (runtime behavior, changed real-time characteristics, lines of code) and the benefit gained (fewer operator interactions).

However, most of these approaches don't take agents as the underlying paradigm into account but focus on the capabilities of the production system as a whole. Therefore, we need to investigate and define metrics that do consider agents as an integral part of the CPPS and derive ways to measure solutions (practices) that utilize them. IEEE P2660.1 working group has attempted to address exactly this issue, and although the focus is constrained on the interface between agent systems and low-level automated devices [26], the results up to now are generalizable to agent-based CPPS. For instance ISO 25010 [37] proposes several criteria for software systems, that could be relevant for agent-based CPPS [27]. Also ISO 25023 [38] has defined explicit measures that could be used to assess a software system, and many of them (but not all) are seen as a good fit for agent-based CPPS [28]. However, although metrics defined in existing standards can be reused, there is a need to find the white spots and propose either new metrics that capture the interplay between agents and CPPS, or enhance existing standard metrics to effectively measure the key aspects of agent-based CPPS, including CPPS system of systems [23].

D. Distributed Intelligence & Machine Learning

Intelligence has always been in the scope of Industrial Agents, especially when they need to handle dynamic situations and adjust to their environment. The last years, rapid advances in the availability of high-performance hardware, big data, and algorithms that can capitalize on both, have given a significant push to machine learning practical applications. As an example, deep learning [39] utilizes neural networks in conjunction with GPUs and can result in task-specific performance that is superior to the human one e.g. image-based classification. This has significant real-world applications, e.g. in quality control in industrial product lines [40].

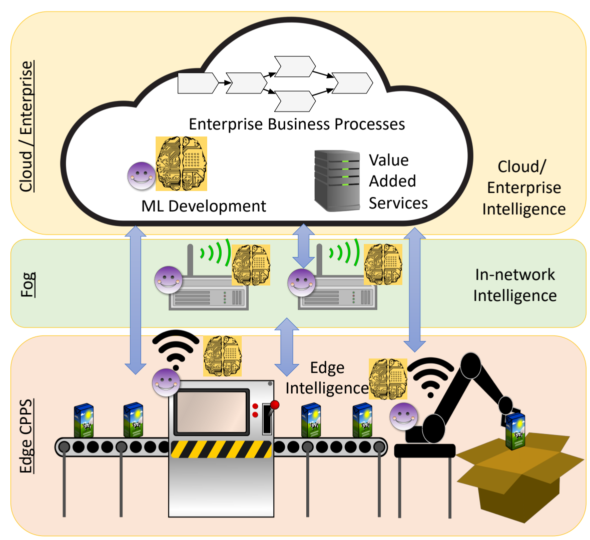

Figure 3. Distributed Intelligence at Cloud, Fog and Edge Levels

As shown in Figure 3, intelligence can be utilized at the cloud, and this is preferred for intensive computational tasks, e.g., during training in machine learning. However, with the emergence of cheap lightweight but highly capable hardware that can be used in with edge devices, this intelligence can nowadays be also deployed at the fog/edge, and this can be lead to a paradigm shift on the way applications and processes are designed, developed and operated in the whole

Preprint version of doi:10.1109/icphys.2019.8780360

factory. For instance, while training can be done on the computationally-rich cloud, the inference can be carried out on the fog/edge by utilizing cost-effective hardware such as Intel's Movidius Neural Compute Stick or Nvidia's Jetson. With such customized hardware that can run efficiently machine learning algorithms, several questions arise e.g. how much intelligence could/should be hosted on edge devices and how much in the fog or cloud, how collaboration at device level can lead to more autonomous and sophisticated systems, if not only inference but also transfer learning can be realized on edge etc. All of these have the potential to lead to new CPPS system designs, as traditional models where data had to be outsourced to more powerful in-cloud infrastructure may no longer be necessary and where inference or transfer learning can be done on the edge and in a collaborative mode.

For CPPS this opens the prospect of highly dynamic and evolving systems that driven by intelligent industrial agents can be potentially better and autonomously handle non-deterministic events, while also empowering emergent behaviors at the infrastructure level. The role of agents needs to be further investigated and see if they can offer advantages when compared to more traditional (non-agent based) approaches. One direction is that agents can adopt an enabling role, facilitating the management, deployment and configuration of the distributed intelligent processes that will run on the cloud, fog, and edge, while being reconfigurable and deployable at runtime. However, they may also be active participants, where agent technology is combined with machine learning, eventually capitalizing on both worlds. For instance, the negotiation capabilities of the agents could be coupled with the machine learning algorithms for image classification (e.g., implemented within the agent), and utilized in a production line to monitor the visual quality of the products. Agents could represent algorithms with different levels of performance that compete for efficient utilization of available computing capacity.

IV. Discussion

Design patterns are a fundamental tool to empower the development of CPPS. However, CPPS design patterns are more complex than traditional automation patterns, as they occur at different design stages and must be aligned across these stages to effectively generate value. This development and alignment are not linear since a plethora of functional and non-functional requirements arises at each stage. Patterns satisfying one set of requirements must align among themselves to create an overall cohesive design. Without such cohesion, it becomes virtually impossible to asses and further operate a CPPS without a considerable re-engineering effort even for smaller changes.

Design patterns rely almost entirely on the specification of several interfaces across many different technical dimensions. Interoperability is a key issue when considering this multilevel interfacing, requiring new scientific and technological solutions to implement transparent and compliant interfaces. The developed interface solutions should be analyzed under two perspectives: i) interconnection among agents, which requires an extension and/or tuning of the FIPA specifications, and ii) between cyber and physical counterparts which requires recommendation practices that better fits the application particularities. In both cases, it is important to consider the use of non-proprietary technologies and standards, e.g., OPC-UA, which is currently widely adopted by industry, or IoT technologies like MQTT (Message Queuing Telemetry Transport) that is based on publish/subscribe paradigm. Also, the adoption of open and standardized approaches for data exchange and portability included in these interfaces, e.g., AutomationML or FIREWARE data models, is strongly recommended.

In the Industry 4.0 context, the alignment of these interface practices with the RAMI 4.0 reference architecture is crucial. As shown in Figure 4, I4.0 components encompass of the assets and an administration shell that abstracts their capabilities and enables their interconnection in the RAMI. Agents could be used to implement the asset administration shell (AAS) or support it by providing the key functionalities e.g. data gathering, physical object encapsulation, communication, intelligent and autonomous decision-support etc.

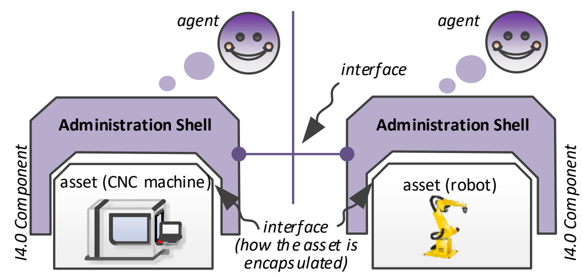

Figure 4. Mapping agents to the Asset Administration Shell (AAS), in order to implement the Industry 4.0 components as defined in RAMI 4.0 [7]

Metrics play a fundamental role in understanding different CPPS instantiations, and can be used to decide upon best ways to tune existing CPPS or realize future CPPS. Especially metrics pertaining to industrial agent-based CPPS are needed and although some have been proposed in the literature, most are not specific for agent-based CPPS. Standards exist such as ISO 25010 [37] that proposes several criteria for software systems, that could be relevant for agent-based CPPS [27]. Having a look at the concrete metrics for those criteria e.g. as provided by ISO 25023 [38], it is evident that although many of them could be adopted, there is still a need to specify new metrics that better capture the interplay of agents and CPPS [28]. As such, research accompanied by an empirical evaluation of existing and prospective implementations of agent-based CPPS is needed.

A well-defined set of metrics and quantitative models are preconditions for machine learning and AI algorithms to operate effectively in the CPPS context. Machine learning has emerged in the last years due to the significant computational and hardware capabilities available, also at the edge. As such, intelligent agents could see a renaissance, and utilize state of the art machine learning to empower their decision making processes at the edge. This has the potential to lead to new innovative solutions, that could rely on training to happen in the cloud, but having the inference done on the fog or edge level, which is seen as fit for real-time decisions. Although

Preprint version of doi:10.1109/icphys.2019.8780360

the agents could be used as enabling technology to manage intelligent approaches, their inherent characteristics (incl. autonomy, negotiation, mobility, etc.) might be more beneficial when combined with distributed intelligence approaches and lead to better services and applications on edge.

V. Conclusions and Outlook

The realization of industrial agent-based CPPS in order to address the needs of modern production systems can be attained once several key challenges are effectively addressed. Four promising directions which may yield the necessary results are identified i.e. design patterns, interfaces, metrics, and distributed intelligence. The next steps comprise of investigating and addressing the white spots in these four directions, and more specifically to better understand the agent-based CPPS patterns, adopt and/or create suitable interfaces that capture sufficiently their interactions, propose metrics to accurately measure behaviors of implementations and practices, as well as capitalize on the advances of distributed intelligence (between edge, fog, and cloud). Such efforts should go beyond simple constellations and include large-scale systems of CPPS. While such work does not start from scratch, the requirements set by the industrial systems and their operational context, make such endeavor challenging.

References

[1] 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*. Springer, 2014.

[2] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP*, vol. 17, pp. 9–13, 2014.

[3] P. Leitão and S. Karnouskos, *Industrial Agents: Emerging Applications of Software Agents in Industry*, 1st ed. Elsevier, 2015.

[4] 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.

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

[6] 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, Mar. 2017.

[7] P. Adolphs *et al.*, "Reference Architecture Model Industrie 4.0 (RAMI4.0)," ZWEI, VDI/VDE, Tech. Rep., 2015. [Online]. Available: https://goo.gl/3DcvkQ

[8] S.-W. Lin *et al.*, "The Industrial Internet of Things – Volume G1: Reference Architecture," Industrial Internet of Things Consortium (IIoT), Tech. Rep., 2017. [Online]. Available: https://goo.gl/5sa56a

[9] "Foundation for Intelligent Physical Agents (FIPA)." [Online]. Available: http://www.fipa.org

[10] M. Wooldridge, *Introduction to Multi-Agent Systems*. John Wiley & Sons, 2002.

[11] R. Unland, "Industrial agents," in *Industrial Agents*. Elsevier, 2015, pp. 23–44.

[12] 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.

[13] A. Giret and V. Botti, "Holons and agents," *Journal of Intelligent Manufacturing*, vol. 15, no. 5, pp. 645–659, 2004.

[14] L. Ribeiro and M. Hochwallner, "On the design complexity of cyber-physical production systems," *Complexity*, vol. 2018, pp. 1–13, Jun. 2018.

[15] A. M. Farid and L. Ribeiro, "An axiomatic design of a multiagent reconfigurable mechatronic system architecture," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1142–1155, Oct. 2015.

[16] V. Marik and D. McFarlane, "Industrial adoption of agent-based technologies," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 27–35, 2005.

[17] A. Lüder, A. Cala, J. Zawisza, and R. Rosendahl, "Design pattern for agent based production system control – a survey," in *IEEE CASE*, 2017.

[18] B. Vogel-Heuser, *Softwareagenten in der Industrie 4.0*. Walter de Gruyter, 2018.

[19] P. Göhner, Ed., *Agentensysteme in der Automatisierungstechnik*. Springer, 2013.

[20] VDI/VDE 2653, "Multi-agent systems in industrial automation," 2018, parts 1–3.

[21] L. Ribeiro and P. Linder, "Hardware abstraction layer for java-based agents," in *IEEE IECON*. IEEE, 2016, pp. 4896–4901.

[22] D. Schütz, A. Wannagat, C. Legat, and B. Vogel-Heuser, "Development of PLC-based software for increasing the dependability of production automation systems," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2397–2406, Nov. 2013.

[23] A. W. Colombo, S. Karnouskos, and T. Bangemann, "A system of systems view on collaborative industrial automation," in *IEEE ICIT*, 2013.

[24] I. Seixas and P. Leitão, "Standards Compliance in Industrial Agents Applications," in *IEEE IECON*, Nov. 2013.

[25] L. M. Camarinha-Matos, R. Fornasiero, and H. Afsarmanesh, "Collaborative networks as a core enabler of industry 4.0," in *Collaboration in a Data-Rich World*. Springer, 2017, vol. 506, pp. 3–17.

[26] P. Leitão, S. Karnouskos, L. Ribeiro, P. Moutis, J. Barbosa, and T. Strasser, "Integration patterns for interfacing software agents with industrial automation systems," in *IEEE IECON*, Oct. 2018.

[27] S. Karnouskos, R. Sinha, P. Leitão, L. Ribeiro, and T. Strasser, "Assessing the integration of software agents and industrial automation systems with ISO/IEC 25010," in *IEEE INDIN*, Jul. 2018.

[28] ——, "The applicability of ISO/IEC 25023 measures to the integration of agents and automation systems," in *IEEE IECON*, Oct. 2018.

[29] M. Weyrich, M. Klein, J.-P. Schmidt, N. Jazdi, K. D. Bettenhausen, F. Buschmann, C. Rubner, M. Pirker, and K. Wurm, *Evaluation Model for Assessment of Cyber-Physical Production Systems*. Springer, Oct. 2016, pp. 169–199.

[30] K. Samir, M. Khabbazi, A. Maffei, and M. A. Onori, "Key performance indicators in cyber-physical production systems," *Procedia CIRP*, vol. 72, pp. 498–502, 2018.

[31] A. Lüder, B. Vogel-Heuser, N. Schmidt, and J. Prieler, "Metric based modelling of flexibility properties of demonstration plants," in *IEEE INDIN*, 2017.

[32] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621–641, 2016.

[33] J. Ladiges, A. Fay, C. Haubeck, and W. Lamersdorf, "Operationalized definitions of non-functional requirements on automated production facilities to measure evolution effects with an automation system," in *IEEE ETFA*, Sep. 2013.

[34] P. Marks, X. L. Hoang, M. Weyrich, and A. Fay, "A systematic approach for supporting the adaptation process of discrete manufacturing machines," *Research in Engineering Design*, vol. 29, no. 4, pp. 621–641, Aug. 2018.

[35] C. Raibulet, "Facets of adaptivity," in *Software Architecture*, R. Morrison, D. Balasubramaniam, and K. Falkner, Eds. Springer, 2008, pp. 342–345.

[36] B. Vogel-Heuser, S. Rösch, J. Fischer, T. Simon, S. Ulewicz, and J. Folmer, "Fault handling in PLC-based industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation," *Journal of Software Engineering and Applications*, vol. 09, no. 01, pp. 1–43, 2016.

[37] "ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models," 2011.

[38] "ISO/IEC 25023:2016 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality," 2016.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[40] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 315–322, May 2018.

Preprint version of doi:10.1109/icphys.2019.8780360