

# Preliminary Insides for an Anthropocentric Cyber-physical Reference Architecture of the Smart Factory

Constantin B. ZAMFIRESCU<sup>1</sup>, Bogdan C-tin PIRVU<sup>1</sup>, Jochen SCHLICK<sup>1</sup>, Detlef ZUEHLKE<sup>1</sup>

DFKI - German Research Center for Artificial Intelligence, Innovative Factory Systems,  
Trippstadter Strasse 122, Kaiserslautern, Germany,  
Bala-Constantin.Zamfirescu@dfki.de,  
Bogdan-Constantin.Pirvu@dfki.de,  
Jochen.Schlick@dfki.de,  
Detlef.Zuehlke@dfki.de

**Abstract:** The classical view of cyber-physical systems is that the integration of computing, communication and control elements are considering only the physical and computational elements, neglecting the human one. This paper presents a view on an anthropocentric cyber-physical reference architecture for smart factories (ACPA4SF), where the key characteristic of its reference model relies on its unified integrality which can not be further decomposed into smaller engineering artefacts without losing its functionality. The paper describes some preliminary insides in this research direction, namely the reference model of the ACPA4SF, its composite types and the enabling technological approaches.

**Keywords:** factory automation, cyber-physical systems, reference architecture

## 1. Introduction

In the last decades automation gains became sufficiently advanced to consider that all things should be connected in the attempt of achieving any significant improvement. Unlike the traditional embedded systems that emphasize the closed control-loop between the computational and physical components, cyber-physical systems (CPSs) extend the computational capabilities to the interaction with similar networked components. It basically adds the feature of integrated sociability within a heterogeneous environment (i.e. computational, physical and social), and brings a paradigm shift in engineering these systems: from computing as algorithm to computing as interaction. The smart factory concept transfers this paradigm to the domain of factory automation [1].

A CPS is usually defined, in terms of its core characteristics that differentiate it from the conventional systems (i.e. embedded systems, real-time systems, sensor networks or desktop applications), such as [2][3]: *integral* (its functionality is relying on the tight integration of its composite elements with self-organization capabilities), *sociable* (the ability to interact with other CPSs via different communication technologies in an open mixed network environment), *local* (the cyber and physical capabilities of a CPS are bounded by the spatial properties of the environment), *irreversible* (self-referential timescale, sensed as dynamics, not discrete, nor spatial), *adaptive* (with self-organization capabilities, such as

learning, adaptation, auto-assembly, etc.), *autonomous* (control loop must close), and *highly automated* (as a key driving-force of eroding the boundaries between its composite elements). All of these offer not only unlimited opportunities for the effective optimization of the production and its support processes (i.e. maintenance, quality), but an engineering abstraction for coping with the complexity of factory automation characterized by decentralization, conflicting requirements, continuous evolution and deployment, and emergent behaviours.

Nevertheless the classical view of CPS is that the integration of computing, communication and control elements are considering only the physical and computational elements, neglecting the human one. Although the trend in CPS research is to rely less and less on human intervention and more and more on the "intelligence" of automated elements, it is obvious that as long as there is no unifying theory of heterogeneous control and communication systems many problems concerning CPS will remain undecidable [4]. Consequently, in any industrial automation the final decision will belong to humans as the ultimate element of the decision chain [5]. This anthropocentric view over CPSs is well supported by the cybernetics' Law of Requisite Variety [6] which states that for any controlled system, the controller of that system must have the aptitude to grasp all possible inputs that may affect the system. In fact this comprehensive view of CPSs is acknowledged by the National Institute for Standards and

Technology [5] that envisions a networked, cooperating, human-interactive systems able to amplify the aptitude of human operations (physical or cognitive) through high levels of situation awareness and adaptability. Moreover, there is clear evidence that computational and physical elements may not be engineered in isolation to each other [7] and requires human intervention to support the cyber-physical intelligence [8]. That is way we defined the reference model for the smart factory as an anthropocentric cyber-physical system (ACPS) which is further used as an abstract template to guide the development of ACPA4SF.

Besides technological developments of industrial communication and industrial automation a common architecture is most important to make the vision of smart factory come true. According to our knowledge, there is no reference architecture for factory automation as a shared baseline of why, what and how to engineer a CPS-based smart factory. The paper describes some preliminary insides in this research direction. Consequently, the next section will define the ACPS reference model in terms of its composite entities and core relationships among them. The basic ACPS types that compose the reference architecture are identified and described in the third section. The fourth section will summarize the enabling technological approaches for instantiating the proposed ACPA4SF. The

paper concludes with some remarks regarding the engineering issues and future work in this direction of factory automation.

## 2. ACPA4SF Reference Model

According to OASIS [9], a reference model defines in an abstract way the core relationships among the composite entities to guide the development of a referential architecture. Consequently, Figure 1 depicts in an UML composite structure diagram the relationships between the three constituents of an ACPS: the physical component (PC), the computational/cyber component (CC) and the human component (HC). The interactions between these components are usually made via adaptors (in many cases, optional) that translate the signals into the specific format of the interacting component. For example, between the PC and HC there are special displays or meters to measure the working parameters of a machine, between the CC and HC there are the classical human-computer interaction (HCI) devices, such as screens, mouse, keyboard etc., while between the PC and CC there are special transducers or analog-to-digital converters (ADC).

These components are connected outside the ACPS to their specific dimension: physical (e.g. via mechanical slots), computational (e.g. via computer-specific communication

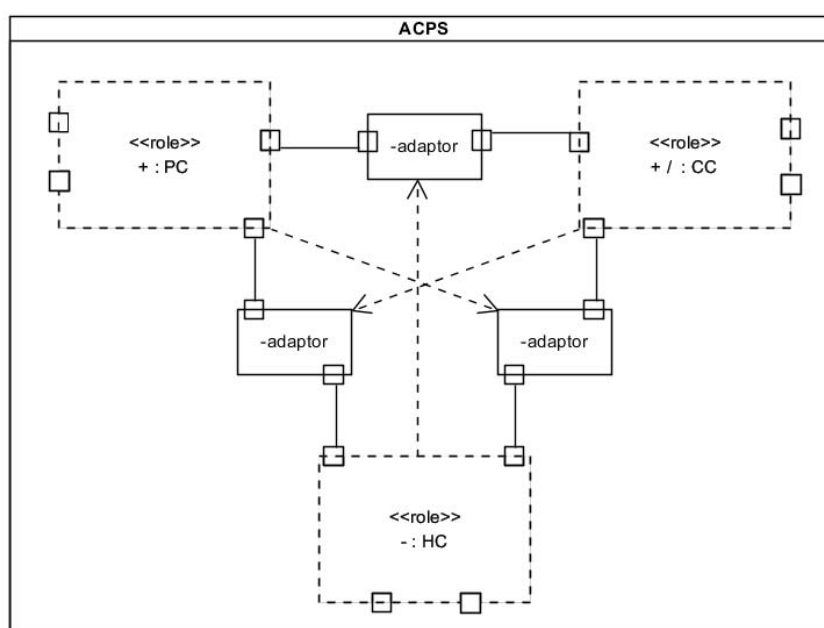


Figure 1. ACPS structure

standards), and social (e.g. via natural language). Note that all these components participate on a role-basis in an ACPS, any component being in an indefinite number of possible ACPSs. A role basically implements an interaction protocol (or a method). In factory automation there are defined several standardized interaction protocols which can be reused or redesigned in a concrete ACPA4SF. Thus, when a new ACPS is designed from an existing one, it may potentially inherit all the interaction protocols (e.g. between PC-HC, HC-CC, or PC-CC). In this way the mechanisms, that previously proved to be useful in coordinating the inner components of an ACPS, can be reused in engineering the new ones. As in any chaotic system, inheriting the interaction protocols does not guarantee the same behaviour of the ACPS as long it depends on the exogenous factors where it is embedded (e.g. initial conditions, external multi-dimensional signals that are account for, etc.) and the multiple adaptation loops of:

- *PC-HC interaction*; in today factories humans must supervise an automated systems or closely co-work with them in a “men-in-the-mesh” manner. That is way the broad topic of human-machine interaction has a long multidisciplinary research track record that cover a broad range of topics [10], such as human-machine interface, human-machine interaction, ergonomics and usability from multiple perspectives (i.e. physical, cognitive, neuro-physiological, social, etc.). All these studies provide clear evidence of the continuous mutual adaptation loop between HC and PC in the manufacturing process.
- *HC-CC interaction*; beyond the classical approaches from the HCI research domain, in the last decade a special emphasize is given on intelligent user interfaces that incorporate adaptive capabilities to personalize and guide the interaction with the humans for better performance or usability [11]. Even in factory automation, the requirement for adaptive capabilities is intensely considered in the new research topic of “adaptive automation” where new concepts, such as neuroergonomic or brain-computer interfaces are frequently used. Nevertheless, these research results argue for an adaptive and active allocation of

tasks between human and computer to provide an ideal workload balance (e.g. [12][13]).

- *PC-CC interaction*; if we carefully look into the latest standardization effort to design low level control systems (i.e. [14]), they are clearly setting the framework for introducing adaptive and intelligent control methods in factory automation. While “adaptive control” and “intelligent control” are well-established research fields in factory automation, they are mostly based on differential equation-based models of the controlled systems. The new “protocol-based” standards, such as MTConnect [14], open the doors for agent-based models with certain advantages in engineering an ACPS, a domain characterized by a high degree of localization, uncertainty, distribution and dominated by discrete decisions.

As may be observed the key characteristic of an ACPS reference model relays is its unified integrality which can not be further decomposed into smaller engineering artefacts without loosing its functionality. Consequently, as long as an ACPS does not pose any HC/PC/CC boundary it becomes the basic abstract building for defining the ACPA4SF. This assumption is well argued in many research fields that inevitably are dealing with the relationships between the composite entities in a unified way:

- *CC-PC relationship*. From a multidisciplinary perspective, the integral relationship between the PC and CC has been studied extensively in robotics, rejecting the traditional assumption of dualism that sees matter and mind/control as independent constituents of reality. For instance in [7] are many examples demonstrating that the behaviour of any system is not merely the outcome of an internal control structure, but it is also affected by the environment in which the system is physically embedded, by its morphology, and by the material properties of the composite elements.
- *HC-CC relationship*. In information science the most widely used research framework for longitudinal studies is the adaptive structuration theory [15]. Basically, the theory criticizes the

technocentric view of using a technology and stresses the anthropocentric aspect, by observing that humans who are using a technology for their work create individual perception about its role. This perception is very dissimilar across groups and has a great influence on how the technology is further used.

- *HC-PC relationship.* The cognitive science gives many clear-cut arguments that the boundary between the different areas of the brain, the brain and body, and the body and environment is indeterminate [16]. Moreover, distributed cognition argues that the acquisition, propagation and processing of information is a distributed process that always happen across a network of humans and artefacts [17]. Consequently, human cognition is situated and embodied, with the power to detect fine-grained patterns and correlations between millions of multi-dimensional signals that are impossible or impractical to transfer into computational systems. Given this reality, even from sixties Licklider [18] envisioned the “man-computer symbiosis” to perform a certain task, until the currently growing research filed of “human computation” [19]. Note that in this later case, the computer application is functioning as a mediator for human intervention over the physical world, whilst the opposite is also true given the complementary aptitude of the two components.

### 3. ACPS Types

The basic ACPS types for factory automation are well reflected in the decentralized approaches for factory automation (i.e. multi-agent, holonic, service-oriented, etc.) where the decomposition is following as accurate as possible the physical reality and not the desired functionalities that may emerge from many interacting components. Consequently, ACPA4SF is a composition of four ACPS types (Figure 2): 1) the *production system* (includes the production resources available in the factory, such as machines, transportation and storage); 2) the *product design system* (includes all the necessary production knowledge to manufacture a product, such as the aerodynamic design of a car or its manufacturing operations flow); the *planning and control system* (include the orders from the customers in terms of product instances in the factory automation framework); and the *infrastructure system* (include the contextual data that is not accessible or foreseen by the previous components but required by them to be operational, such as buildings, rooms, technological infrastructure).

These basic types are self-sufficient to describe any factory and are answering the key questions regarding the engineering of a manufacturing control system: *product design system - what products will be manufactured and how can they be manufactured?* (N.B., the

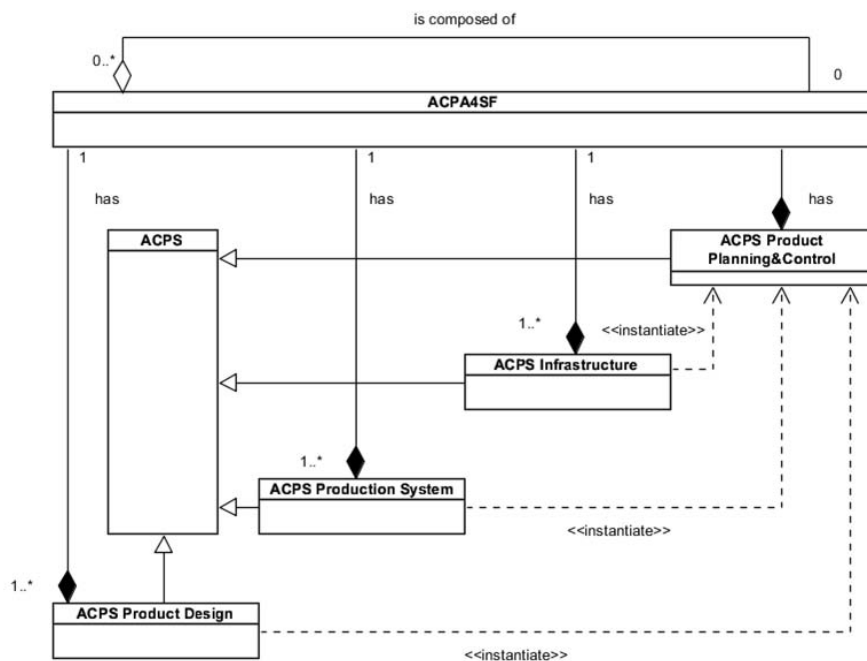


Figure 2. Basic ACPS types used in factory automation

answer to this question is a production plan for a product type, not a recipe that needs in addition to allocate certain resources to realize the plan); production system - *what, how and when a resource can be used?* (N.B., usually the production equipment in a factory has a certain degree of autonomy in deciding what and when a manufacturing operation will be executed); planning and control system – *which is the most appropriate execution flow of operations to manufacture a product instance?* (N.B., it includes also the non-operational steps of the manufacturing process, such as transportation, storage, etc.); infrastructure system - *which are the physical, real-world, constraints of the manufacturing environment?* (N.B. modularity and flexibility in designing and reconfiguring the plant layout play a crucial role as regards the key performance indicators of the factory).

### 3.1 Product design

Today's product design within the manufacturing domain implies a complex set of tasks supported by various software applications and physical devices from the product's conceptual design until its embodiment. During this process the tight interactions among the main components of ACPS are evident:

- *HC-CC interaction*: personalization capabilities of the current CAx-tool environment to meet the specific user needs (e.g. changing the tool bar or the 3D visualization characteristics) are virtually pervasive; moreover the selection of the appropriate simulation tool from the entire CAx-suite to tackle a specific product problem (e.g. simulation of the product's rigidity – CAE, simulation of the assembly process – Digital Mock-up, etc.) is seamlessly guided by the modern available product design tools;
- *HC-PC interaction*: involves classical ergonomics and usability assessment of the product prototype, as commented in the previous section;
- *CC-PC interaction*: includes for example scanning the clay model into the CAD-environment, rapid prototyping the CAD model on a 3D printer, part milling on a CNC-Machine.

### 3.2 Production system

The production system of any factory is composed of three basic types: processing, storage, and transportation. There is also possible to have hybrid resources: equipment with the double/triple function of processing and transportation and/or storing at the same time. Moreover, production resources may be aggregated (i.e. workstations, work-cells, etc.) or specialized depending on the desired level of granularity in the implementation of a concrete ACPA4SF. It can be done at the component level (cyber, physical, social) or at the ACPS level, when besides the functionalities of the composite components the coordination mechanisms (i.e. interaction protocols) may be aggregated and specialized as well, either for direct (as shown in [20]) or indirect ones (as shown in [21]).

The compliance of any type of production system with the ACPS reference model is the most evident one. The interactions between its three main components have been described in the previous section.

### 3.3 Infrastructure

In line with the studies from the research field of multi-agent systems (i.e. [22]), the infrastructure is considered an active ACPS with specific responsibilities: provides a medium for sharing information and mediating coordination among all the ACPSs that compose the ACPA4SF. Consequently, the infrastructure captures all the necessary design issues that influence the behaviours of the others ACPSs types (i.e. product design, production system, and control). For example the plant layout, the network infrastructure (i.e. cable, wireless, etc.), the computing power available in the factory (i.e. servers, high-performance computers, PCs, tablets, smart phones, etc.), the social communication medium (i.e. phone, face-to-face, scoreboard, etc.) may have a great impact over the design choices of engineering a smart factory. Obviously, an exhaustive model of the entire infrastructure is unfeasible, but generally it should capture the key issues which mediate the interactions among the ACPSs types and their access to resources (i.e. physical, computational, communicational, and social).

The view of infrastructure as an ACPS is related to the general research field of

computer-augmented environments, with its many subfields, such as ubiquitous computing, pervasive computing, ambient computing, cooperative buildings/roomware [23], spatial computing, augmented reality, etc. In this domain, the researchers are not only investigating the possibility to control the physical infrastructure of the building (i.e. heating, ventilation, doors and access control) via a network of computers, sensors and actuators, but also the social communication (i.e. audio and video content) or to disclose environmental data to humans on mobile devices (i.e. information about activities in the building, how to find your way in the building, etc.). Moreover, some of them are talking even about dynamic architecture and adaptive buildings [24].

### 3.4 Product planning and control

The product planning and control ACPS type may be found in literature under a diverse spectrum of terms (i.e. order, intelligent product, product instance, planning, scheduling, etc.). Conceptually, it poses some problems in separating its concern, due to the need to exploit knowledge from all the ACPS types. In common cases it takes an aggregated form (i.e. the classical ERP system) where the boundaries between the ACPS types are difficult to distinguish. On the other hand, in the decentralized implementations its distinctive role is practically obvious.

Theoretically it corresponds to the real instantiation and subsequent use of a product

(from raw material to dismantling), but practically the life-cycle of a product instance is limited to the manufacturing stage. Consequently, the ACPS product planning and control may be viewed as the work-piece (i.e. intelligent product) with certain control behaviour to manage its itinerary through the factory by negotiating with other ACPS types to get produced (that is way in Figure 2 the ACPS product planning and control embeds instantiations from the others types). These interactions involve negotiations and exchange of knowledge among the ACPS types (Figure 3). For example, it needs to know: from the ACPS product design how to manufacture the product instance (“product manufacturing knowledge”), from the ACPS production system where and when to execute the processing operations (“process execution knowledge”), and from the ACPS infrastructure if the identified processing resources are reachable at reasonable costs (“context execution knowledge”). In the same way the ACPS product design needs to know: from the ACPS production system which are the possible manufacturing operations available in the plant (“resource-specific product manufacturing knowledge”) and from the ACPS infrastructure in what context their availability is valid (“context-specific product manufacturing knowledge”).

Note that all these knowledge and negotiation activities are happening in a three-dimensional space (physical, computational and human). Consequently, they should not be considered as complete-automated activities, significant parts

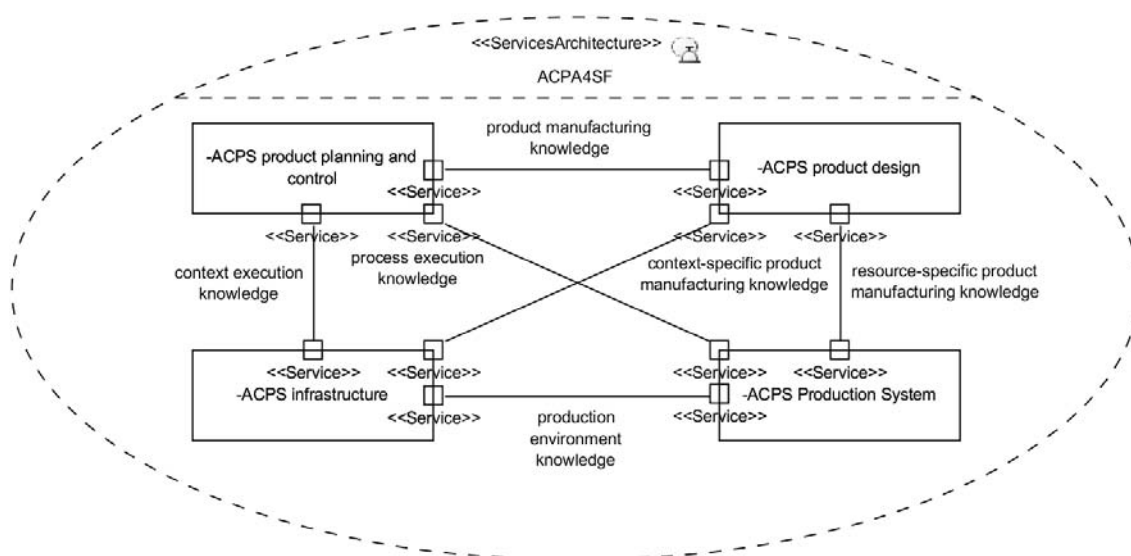


Figure 3. ACPA4SF as a composite structure diagram of ACPS types

being realized via social (when the operators or engineers are directly coordinating their activities in face to face meetings) or physical (when an intermediate storage buffer is overloaded) communication channels. Therefore the services represented in Figure 3 are aggregated services that comprise all possible services provided by the ACPS reference model.

## 4. Enabling Technological Approaches

The enablers for instantiating the proposed ACPA4SF are widely recognized in the research community of factory automation. These are the: the agent-oriented (AO) approaches (including its subsequent developments, such as holonic and product-driven manufacturing systems), service-oriented architectures (SOA) and ontologies. Even if they are considered dissimilar research domains, in practical applications they are unavoidable merged in a unified technology. The ontologies are used in both AO and SOA approaches to enable semantic interoperability, while AO and SOA are considering complementary issues of engineering intelligent distributed systems.

### 4.1 Agent-based technologies

With over three decades of research history, AO design is a software engineering paradigm where the primary decomposition unit is an autonomous entity (i.e. agent [25]) that enjoys very similar properties with a CPS, such as autonomy, locality, adaptation, cooperation, trustworthiness, etc. As long there is no any other software engineering abstraction able to reflect the distributed control nature of an ACPA4SM, the implementation of an ACPS should strongly rely on its achievements. That is why most concepts and standards that are used nowadays to engineer the CPSs are influenced by the principles of AO design.

During its already long history, the AO approach significantly contributed with a complete set of methodologies and tools that cover any aspect of engineering agent-based systems. In addition, it brings significant theoretical insides as regards the design patterns of interaction protocols and the relevant structure for high-level communication languages among agents. Due to its inherent cognitive complexity, unfortunately all these

achievements have been adopted by industry at a very limited scale [26]. Consequently, AO approach is considered by industry to be a research-driven, leapfrog approach of moving from a centralized architecture to a decentralized one.

### 4.2 Service-oriented architectures

Unlike agent technology, SOA is already a deploying standard for most enterprise integration software providers (e.g. SAP, Microsoft, Oracle, etc). It provides a set of relevant features that promote reuse, growth and interoperability of services (i.e. aggregation, specialization, and generalization). Unfortunately, these implementations are still far away from its initial theoretical promises (i.e. autonomy, flexibility, cooperation, etc. of services). That is why it is widely accepted that agent technology is the key enabler to support the engineering of a true SOA [27]. In fact, most SOAs (e.g. OMG's SoaML, OASIS' Reference Architecture [9], MTConnect [14], etc.) are explicitly using the agent concept, but unlike AO approach, SOA is taking an industry-driven incremental approach of moving from a centralized to a decentralized architecture.

For industrial automation, the common mechanism is to provide services on different scales of aggregation across the factory: the factory components supplies a set of services (from the device level to the application level), and through an appropriate interaction protocols, the right combination of services is selected and synchronized to attain a certain goal. These issues are well reflected in the continuous and gradual improvement process of SOA in many EU projects, such as [28]: SIRENA, SOCRADES, SODA, and ESCOP.

### 4.3 Ontologies

Ontology plays the crucial role of minimizing the collaboration frictions by narrowing the semantic gap between humans working on different automation aspects of the production system. As an immediate consequence there is the possibility for an increased degree of delegation of responsibilities to their artefacts (i.e. autonomous components) which can interoperate if they are using a common agreed formal structure to represent collective knowledge. That is why ontologies are a common ingredient in AO and SOA

approaches for factory automation and in most industrial applications their structure reflects the knowledge associated with the ACPS types described in the previous section (e.g. [29]). Unfortunately, there is still no universal ontology, all the ontologies that have been developed until now exhibiting clear particularities that reflect the common knowledge and experience from the implementation context. Nevertheless, the intensification of communication channels across many boundaries (i.e. organizational, cultural, social, experiential, etc.) will make possible to rely on more general and complete ontologies for factory automation.

## 5. Concluding Remarks

The classical view of CPS is that the integration of computing, communication and control elements are considering only the physical and computational elements, neglecting the human one. In this paper we have argued for an integrated approach as regards the basic abstract building block for defining the ACPA4SF, namely the ACPS reference model. On this basis, ACPA4SF has been defined as a composition of four ACPS types (i.e. product design, production system, infrastructure, and planning and control) that are self-sufficient to describe any factory and are answering the key questions as regarding the engineering of a manufacturing control system. The enabling technological approaches for instantiating the proposed ACPA4SF have been summarized as well.

If the tight combination of physical, computational and social components is almost evident in the “men-in-the-mesh” operational context, for others circumstances the proposed ACPA4SF has manifold implications which clearly differentiates it in respect to the classical perspective of industrial automation which see the computational element as a wrapper for the physical one:

- the computational component should manifest adaptive properties for the changes of both components: physical (i.e. when the machine’s capabilities are declining in time) and social (i.e. when the human operator is improving his/her skills of operating the machine); as a corollary, the roles of this interplay are not fixed at

the design stage, but are adapting in real-time during the task execution;

- an ACPS is considered over its entire life-cycle, not only in its normal operational phase; consequently, an ACPS is grown, not built, requiring incremental evolution as new requirements will be discovered to be useful or abandon the unnecessary ones;
- the design of the computational, physical and social components are following the concurrent engineering practice, and not the classical flow of engineering an automated system: machine, software, and human operator’s training;
- the engineering of an ACPS should consider the mediated interactions between the composite components (human to computer via machine, and so on); in these circumstances any component play an active role in shaping these interactions;
- an ACPS manifest self-awareness capabilities that incorporate both: context-awareness and inner-awareness; this is a key issue to reflect (mirror) as accurate as possible one dimension of the ACPS (i.e. cyber, physical, human) into the others two; consequently, designing an ACPS is hardware-software-useware continuous co-design, which goes far beyond the traditional approach of industrial automation (i.e. hardware-software co-design);
- an ACPS is the intersection point of the three dimensional cyber-physical-social space, consequently it is the key design element that can provide reflection of one space into the others; the accuracy of this reflection will strongly influence the decisions that are made as regards task allocation between humans and control software.

As mentioned, these ACPS types may be aggregated at different levels of abstractions, depending on the hierarchical structure of a concrete instance of ACPA4SM. Its instantiation for different control patterns (i.e. centralized, hierarchical, and heterarchical) will be investigated in our further work. A first step in this direction is given in [30].



## Acknowledgements

This work is supported by the “CyProS” project, funded by the German Government (<http://www.projekt-cypros.de/>).

## REFERENCES

1. ZUEHLKE, D., **SmartFactory-Towards a Factory-of-things**, Annual Reviews in Control, vol. 34, no. 1, 2010, pp. 129-138.
2. RAJKUMAR, R., **CPS Briefing**, Carnegie Mellon University, 2007.
3. LEE, A. E., **Cyber Physical Systems: Design Challenges**, International Symposium Object / Component / Service-Oriented Real-Time Distributed Computing, 2008, SUA, pp. 363-369.
4. NIST, **Foundations for Innovation in Cyber-Physical Systems**, Workshop Report, 2013, <http://www.nist.gov/el/upload/CPS-WorkshopReport-1-30-13-Final.pdf>
5. CHITUC, C. M., F. J. RESTIVO, **Challenges and Trends in Distributed Manufacturing Systems: Are Wise Engineering Systems the Ultimate Answer?** Second International Symposium on Engineering Systems MIT Press, 2009, Cambridge, Massachusetts.
6. ASHBY, W. R., **Requisite Variety and Its Implications for the Control of Complex Systems**, Cybernetica, vol. 1, no. 2, 1958.
7. PFEIFER, R., J. C. BONGARD, **How the Body Shapes the Way We Think**. A New View of Intelligence, MIT Press, 2007.
8. ZHUGE, H., **Interactive Semantics**, Artificial Intelligence, vol. 174, no. 2, 2010, pp. 190-204.
9. OASIS, **SOA Reference Model**, 2012, <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html>
10. ZÜHLKE, D., **Use-ware-Engineering für technische Systeme**, Springer-Verlag, 2004.
11. SUDUC, A. M., M. BÎZOI, F. G. FILIP, **User Awareness about Information Systems Usability**, Studies in Informatics and Control, vol. 19(2), 2010, pp. 145-152.
12. PARASURAMAN, R., C. D. WICKENS, **Humans: Still Vital after All These Years of Automation**, Human Factors, vol. 50, 2008, pp. 511-52.
13. LAGU, A. V., S. J. LANDRY, H. YOO, **Adaptive Function Allocation Stabilization and a Comparison of Trigger Types and Adaptation Strategies**, International Journal of Industrial Ergonomics, 2013, in press.
14. MTCConnect, <http://www.mtconnect.org/>
15. DESANCTIS, G., M. S. POOLE, **Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory**, Organization Science, vol. 5, no. 2, 1994, pp. 121-147.
16. ANDERSON, M. L., M. J. RICHARDSON, A. CHERO, **Eroding the Boundaries of Cognition: Implications of Embodiment**, Topics in Cognitive Science, 2012, pp. 1-14.
17. HUTCHINS, E., **Cognition in the Wild**, 1995, MIT Press.
18. LICKLIDER, J. C. R., **Man-computer Symbiosis**, IRE Transactions on Human Factors in Electronics, vol. HFE-1, 1960, IEEE pp. 4-11.
19. QUINN, A. J., B. B. BEDERSON, **Human Computation: A Survey and Taxonomy of a Growing Field**, CHI, 2011, Canada.
20. BELLIFEMINE, F. L., G. CAIRE, D. GREENWOOD, **Developing Multi-Agent Systems with JADE**, 2007, Wiley.
21. FERNANDEZ-MARQUEZ, J. L., G. DI MARZO SERUGENDO, S. MONTAGNA, M. VIROLI, J. L. ARCOS, **Description and Composition of Bio-inspired Design Patterns: a Complete Overview**, Natural Computing, vol. 12, no. 1, 2013, pp. 43-67.
22. WEYNS, D., A. OMICINI, J. ODELL, **Environment as a First Class Abstraction in Multiagent Systems**, Autonomous Agent Multi-Agent Systems, vol. 14, 2007, pp. 5-30.
23. STREITZ, N. A., J. GEIBLER, T. HOLMER, **Roomware for Cooperative Buildings: Integrated Design of Architectural Spaces and Information Spaces**, in Cooperative Buildings - Integrating Information, Organization, and Architecture, Springer, 1998, pp. 4-21.

24. GRØNBÆK, K., M. KYNG, P. KROGH, **Intelligent Buildings and Pervasive Computing – Research Perspectives and Discussions**, available at <http://www.cfpc.dk/publications/files/Intelligentbuildings.pdf>
25. WOOLDRIDGE, M., **An Introduction to Multi-Agent Systems**, 2002, Wiley.
26. LEITAO, P., P. VRBA, **Recent Developments and Future Trends of Industrial Agents**, Holonic and Multi-Agent Systems for Manufacturing, LNCS, vol. 6867, 2011, pp. 15-28.
27. CALISTI, M., F. LEYMAN, F. DIGNUM, R. KOWALCZYK, R. UNLAND, **Service-oriented Architectures and Multi-agent Systems Technology**, Dagstuhl Seminar Proceedings 10021, 2010.
28. COLOMBO, A. W., S. KARNOUSKOS, T. BANGEMANN, **A System of Systems View on Collaborative Industrial Automation**, IEEE International Conference on Industrial Technology (ICIT), 2013, pp. 1968 – 1975.
29. KHILWANI, N., J. A. HARDING, A. K. CHOUDHARY, **Semantic Web in Manufacturing**, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, vol. 223, no. 7, 2009, pp. 905-925.
30. PIRVU, B. C., J. SCHLICK, S. HODEK, D. ZUEHLKE, **Conceptual Overview of a Smart-Factory Architecture**, Proceedings of the 6<sup>th</sup> International Conference on Manufacturing Science and Education (MSE), 2013, pp. 179-182.