Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Do Not Cancel My Race with Cyber-Physical Systems

Constantin B. Zamfirescu*,. Bogdan-Constantin Pirvu*. Matthias Loskyll*. Detlef Zuehlke***

* German Research Center for Artificial Intelligence (DFKI), Innovative Factory Systems, Trippstadter Strasse 122, Kaiserslautern, Germany (e-mail: Bala-Constantin.Zamfirescu@dfki.de; Bogdan-Constantin.Pirvu@dfki.de; Matthias.Loskyll@dfki.de; Detlef.Zuehlke@dfki.de).
** Lucian Blaga University of Sibiu, Faculty of Engineering, Department of Computer Science and Automatic Control, Emil Cioran 17, Sibiu, Romania (e-mail:zbc@acm.org)

Abstract: To engineer the factory of the future the paper argues for a reference model that is not necessary restricted to the control component, but integrates the physical and human components as well. This is due to the real need to accommodate the latest achievements in factory automation where the human is not merely playing a simple and clear role inside the control-loop, but is becoming a composite factor in a highly automated system ("man in the mesh"). The concept is demonstrated by instantiating the anthropocentric cyber-physical reference architecture for smart factories (ACPA4SF) in a concrete case study that needs to accommodate the ongoing researches from the SmartFactory[KL] facility (e.g. augmented reality, mobile interaction technology, virtual training of human operators).

1. INTRODUCTION

In the last decade the advances in factory automation became aware of the fact that any significant improvement may be achieved only by considering the tight integration of computational, physical and social elements (NIST, 2013). Today, this comprehensive outlook can be found in very dissimilar research areas (i.e. aerospace, automotive, chemical processes, civil infrastructure, energy, healthcare, manufacturing, transportation, etc.), including for example the IBM's Smarter Planet initiative.

Due to the fast-evolving "intelligence" of the automated systems, the standard view of **cyber-physical systems** (CPSs) is to emphasize the integration of physical and computational elements (Rajkumar, 2007; Lee, 2008), neglecting the essential human's role (Chituc and Restivo, 2009) in solving many of the CPS's undecidable problems (NIST, 2013). If this myopic view works well for simple and decoupled from the real environment problems, it does not offer an adequate engineering abstraction for coping with the complexity of factory automation (i.e. decentralization, conflicting requirements, continuous evolution and deployment, emergent behaviours, etc.).

Consequently, in Zamfirescu et. al. (2013) we defined the **anthropocentric cyber-physical system** (ACPS) as a reference model for factory automation that integrates the **physical component** (PC), the **computational/cyber component** (CC) and the **human component** (HC). The key characteristic of an ACPS reference model is its unified integrality which cannot be further decomposed into smaller engineering artefacts without loosing, due to the relevant interactions, its functionality. This view is well supported by the evidence that computational and physical elements may not be engineered in isolation to each other (Pfeifer and Bongard, 2007) and requires human intervention to support the cyber-physical intelligence (Zhuge, 2010). It is also acknowledged by the National Institute for Standards and Technology (NIST, 2013) in its vision of networked, cooperating, human-interactive systems that are able to amplify the aptitude of human operations (physical or cognitive).

The title of the paper is inspired from the seminal book of Brynjolfsson and McAfee (2011) who advocate the Licklider's (1960) idea of "man-computer symbiosis" as the ultimate way to compete within the cyber-physical world ("to race with the machine and not against the machine"). The idea is exposed in the canonical example of chess games, one of the most competitive analysis laboratories for the race between humans and computers. In a review of "free-style" chess games in which humans and computers could compete in mixed teams, Kasparov (2010) concluded that "weak human + machine + better process is superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process". If this is true for a static and closed-world domain, it should be specifically valid for factory automation which faces a complex, open, uncertain and dynamic environment.

The paper presents a step in this direction by moving from a functional decomposition to an interaction-based architectural design of our SmartFactory[KL] demonstrator. Consequently, the next section will summarize the key elements of our **anthropocentric cyber-physical reference architecture for smart factories** (ACPA4SF). A brief description of the SmartFactory[KL] production system for assembling customizable key-finders is presented in the third section. In the next section it will be detailed as an instantiation of the ACPA4SF. The paper concludes with some remarks regarding the power of our proposed reference architecture to be used as a guideline for integrating the ongoing researches in factory automation from the SmartFactory[KL] facility.

2. ACPA4SF

The ACPS (Figure 1, Zamfirescu et.al, 2013) defines in an abstract way the relationships among its three core composite entities (PC, CC and HC). The interactions between these components are realized via adaptors (in many cases optional) to translate the specific signals into the required format of the interacting component. All these components participate on a role-basis in an ACPS. The main difference of ACPS related to all decentralized approaches for factory automation (i.e. multi-agent, holonic, service-oriented, etc.), is that de atomic decomposition unit is not restricted just to the integration of CC and PC, but integrates the HC as well. This is evident when accommodating the latest achievements in factory automation, because the HC is not just playing a simple and clear role inside the control-loop anymore, but is becoming a composite factor in a highly automated system ("man in the mesh"). In this case the boundaries between PC, CC and HC are less evident. Examples in this regard from our SmartFactory^{KL} facility comprise the employment of seamless augmented reality (Gorecky et.al., 2012), mobile interaction technology in the factory of the future (Schmitt et. al., 2013) and the virtual training of human operators (Stork et. al., 2012).

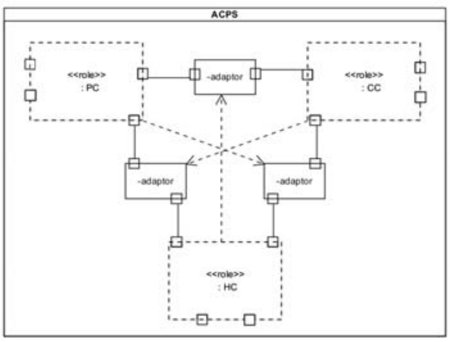

Fig. 1. ACPS structure represented in an UML composite structure diagram.

The ACPA4SF is defined as a composition of four ACPS types which are self-sufficient to describe and engineer any manufacturing control system (Fig. 2, Zamfirescu et. al., 2013): 1) the **ACPS production system** (ACPS-PS) - includes the production resources available in the factory (i.e., machines, transportation and storage); 2) the **ACPS product design** (ACPS-PD) system - includes all the necessary production knowledge and engineering tools to manufacture a product (i.e. manufacturing operations workflow for a product type); 3) the **ACPS planning and control** (ACPS-PC) system - includes the orders from the customers in terms of product instances; and 4) the **ACPS infrastructure** (ACPS-I) system - includes the engineered contextual data and control elements required by the previous ACPS types to operate in a real environment (e.g. buildings, rooms, technological infrastructure). All of these types inherit the core relationships among the composite entities of the ACPS their inclusion in an ACPS instance obviously depends on the engineering compromises that should be accommodated in its real implementation. The ACPA4SF can be viewed as an instantiation of PROSA (Van Brussel et.al., 1998) for smart factories.

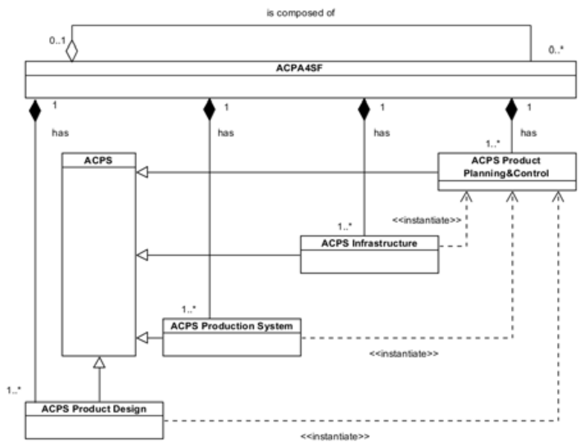

Fig. 2. Basic ACPS types used in factory automation represented in an UML class diagram.

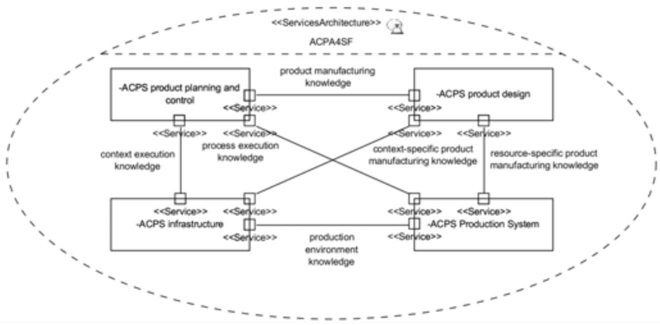

Fig. 3. The interaction among the ACPS types inside ACPA4SF

To get a product instance manufactured there is a continuous interaction flow for exchanging relevant knowledge between the ACPS types (Fig. 3, Zamfirescu et.al, 2013). For example the ACPS-PC that reflects a product instance (i.e. intelligent product) has to manage its itinerary through the factory by negotiating with other ACPS types to get produced (ACPS-PC embeds instantiations from the others types). Consequently, it needs to know: from the ACPS-PD how to manufacture the product instance ("product manufacturing knowledge"), from the ACPS-PS where and when to execute the processing operations ("process execution knowledge"), and from the ACPS-I if the identified processing resources are reachable at reasonable costs ("context execution knowledge"). Similarly, the ACPS-PD needs to know: from the ACPS-PS which are the possible manufacturing operations available in the plant ("resource-specific product manufacturing knowledge") and from the ACPS-I 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 (i.e. physical, computational and social). Consequently, they should not be considered as complete-automated activities, significant parts being realized via social or physical communication channels. Therefore the services represented in Fig. 3 are aggregated services that comprise all possible services provided by an ACPS type.

3. SMARTFACTORY[KL] DEMONSTRATOR

This section describes the production system for assembling customizable key-finders that can be called via smart phones. The line is placed in the SmartFactory[KL] demonstration facility (Zühlke, 2008) and was designed to test advanced paradigms for manufacturing control. The key-finder product includes casting-covers, printed circuit board equipped with LED, loudspeaker and Bluetooth.

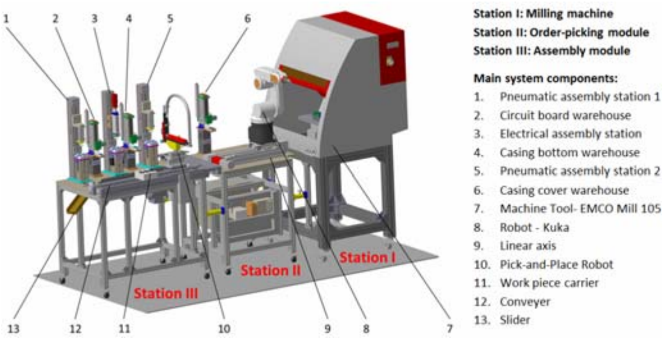

Fig. 4. SmartFactory[KL] production system for assembling customizable key-finders

In the first process step (Fig. 4), the client provides from a control panel some personal preferences as regards both the desired end-product (e.g. name, e-mail, etc. which will be engraved on the key-finder) and/or the manufacturing process (i.e. the shortest delivery time, cheapest production costs in terms of energy consumption, etc.). As in the product intelligence concept (McFarlane et. al., 2013), to increase the flexibility for (possible) changing requirements during the manufacturing process, the implementation is splitting the standard customer order into operational (manufacture-driven) and order (customer-driven) requirements.

In the second phase (Fig. 4), the information from the control panel is transmitted to the second station's PLC, where the Kuka robot picks-up from the warehouse a casing-cover with a RFID-tag glued on it. The robot firstly places the casing-cover over a RFID-Tag writer which stores the abstract processing plan and then places it into the milling machine to engrave the custom information on the casing-cover. At the end of the milling process, the robot takes the casing-cover from the milling machine, places it over the RFID-Writer where the status of the product is written on the RFID-Tag. Finally, after this phase, the casing-cover is passed to the third station by putting the casing-cover to the work-piece carrier.

In the third production phase, the work-piece carrier with the casing-cover goes towards a defined pick-up place from where the pick-and-place robot takes casing-bottoms and circuit boards from their specific warehouses and places them in the available assembly stations. First the casing-bottom is placed in the assembly station, then the circuit board, and afterwards the casing-cover from the work-piece carrier. Finally the assembly station will press these parts together to form the key-finder product. In the last process sequence the pick-and-place robot takes the key-finder and leaves it on the slider, to be picked up by the client.

4. THE INSTANTIATION OF ACPA4SF

This section will detail the instantiation of the ACPA4SF for the SmartFactory[KL] production system for assembling customizable key-finders. For simplicity reasons, even if the ACPA4SF is considering the entire life cycle of an automated factory here, we consider only the operational phase of the current implementation, neglecting its continuous evolution and deployment.

4.1 ACPS-PS

As mentioned, the production system of the key-finder demonstrator is composed of three workstations (i.e. milling, order-picking, and assembling). The first two stations have a classical control architecture executed by a state of the art PLC, while the third one implements a SOA control architecture executed on the microcontrollers that are located within these devices. Despite their dissimilar control architecture they pose a certain degree of autonomy in respect to the manufacturing operations they are providing. The workstations are aggregated production resources, comprising processing (e.g. CNC milling machine, pneumatic assembling stations), transportation (e.g. conveyor belts, robots) and storage (e.g. the warehouses between them) machines.

Worth to mention is that the intermediate warehouses require human intervention (when notified by a warning light signal) to manually fill them with specific sub-components once they are empty. Moreover, any change at the field device level for the first two workstations needs their manual integration into the PLC control. This issue is partially solved for the third workstation that discloses the semantic description of its composite field devices to the others ACPS types of the ACPA4SF in a classical SOA control architecture (Loskyll et al., 2012). Obviously the aggregation level of these services is a problem of control decentralization for the engineered systems. For example, in our case study the pick and place robot (Fig. 4) provides several basic services (i.e. the vacuum and pivoting control and the robot's three-axis control) and an overall service (i.e. for the entire synchronization of the robot's actions) that are executed on different microcontrollers. Even if the latest is implemented in the assembling station it may be equally realized inside the ACPS-PC type.

The semantic description of the field devices is reflected in some special ontology ("device ontology" and "parameter ontology") to capture their inherent characteristics and operational capabilities in terms of either functional (i.e. input and output parameters) or non-functional (i.e. quality of service, consumption level, and context-dependent information) properties (Loskyll et al., 2012). These ontologies are used to support, at the semantic level, the interoperability of ACPS-PS components with the (Fig. 5):

- ACPS-PD to identify the abstract manufacturing operations that can be realized on a specific workstation ("resource-specific product manufacturing knowledge");

- ACPS-I to discover the feasibility of employing the services provided by the workstation and monitor the status of these services in respect to their availability and quality of service ("production environment knowledge");
- ACPS-PC to execute according to the client's requests a certain service ("process execution knowledge").

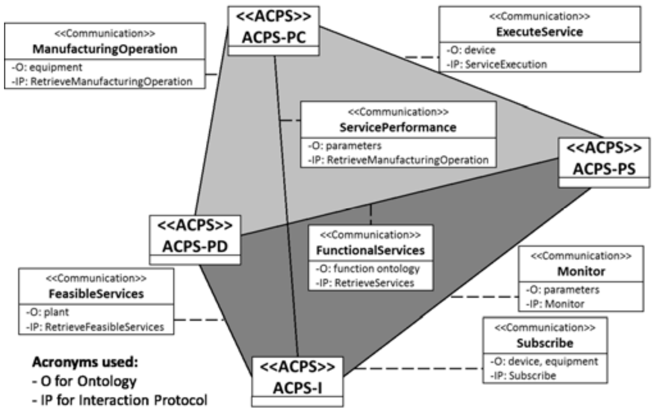

Fig. 5. ACPA4SF society model

4.2 ACPS-I

Basically, the ACPS-I should capture all the necessary design issues that mediate the interactions among the ACPS types and their access to resources. It is an engineered building block, with its own clear-cut responsibilities, irrespective of the others ACPS types (Weyns, Omicini and Odell, 2007). In our knowledge there is no factory without and engineered (n.b. not given) infrastructure that capture the surrounding conditions for the others ACPS types to exist. The ACPA4SF is generalizing all possible compromises to accommodate multiple and conflicting objectives in this respect (i.e. from centralized to decentralized approaches), and not to provide a design guideline for an optimal allocation of responsibilities between the ACPS-I and the others ACPS types. For example in case of a road cross through a dangerous area (i.e. falling stones), is the "infrastructure's" responsibility to close the road when it is required or is the travelers responsibility to take care of the potential danger? In all cases the answer is a cost/benefit analysis of weighting many inconsistent objectives. Obviously, as long as these objectives are not inherent to the real world but constructed by humans (Grubici and Fan 2010), there may be multiple "sources of truth" to reflect them. Our assumption is that a smart factory will provide a semantic-rich communication environment that integrates heterogeneous networks (i.e. social, computational, and physical) and provides a challenging and attractive collaborative working environment for learning, sharing and designing the manufacturing knowledge and objectives. Similar to folksonomies (see Robu, Halpin and Shepherd, 2009), the multiple "source of truth" will emerge, in time, into a truly consistent truth/ontology.

According to Weyns, Omicini and Odell (2007), the ACPS-I may envisage tree levels of support: deployment, abstraction, and interaction mediation. For our case study the deployment context is reflected by the network infrastructure that connects the PLCs from the first two stations, the microcontrollers from the third station and the server. Because the most dynamic and unpredictable deployment context is related to the last station, the ACPS-I builds over it an abstraction level that maintains the states (e.g. running, available or malfunctioning) of the services provided by the ACPS-PS and their level of energy consumption (e.g. low, medium, high). By using a standard SOA architecture, it includes a subscription-based semantic service repository (Loskyll et al., 2011) that facilitates at run-time the discovery and selection of the relevant services provided by the ACPS-PS. In this way any human intervention at the field device-level inside the ACPS-PS assembly module (i.e. their replacement because of component failure or the needs for an improved energy consumption) will be automatically reflected in the ACPS-I which is not the case for the first two stations.

Because the factory demonstrator was not intended as testbed for modular plant automation, the CC of the ACPS infrastructure is limited to monitoring the state of the deployment environment in a predefined layout configuration. The plant layout is captured at the semantic level in an application-specific ontology (i.e. plant ontology) that captures the complex topology (e.g. structural, process-related, physical, electrical) of the devices in the plant (Loskyll et al., 2012). Obviously any change in the layout design of the key-finder demonstrator will require human intervention to refine the "plant ontology", but a slight increase of the level of automation (e.g. by embedding proximity sensors for the automatic sensing of layout changes) will significantly reduce these interventions.

Consequently, the control component of ACPS-I is limited to monitor the ACPS-PS, without the need to support changes of their physical arrangement or to support their interaction mediation. Moreover, this function is not necessary uniform over the entire factory, being implemented only for the last station to accommodate the cost trades-offs of engineering the ACPS-I. All of these left-over parts require a tight interaction between the HC, CC and PC of the ACPS-I.

Table 1. The active role of the ACPS components for the key-finder demonstrator in operational context

ACPS type	PC	HC	CC
ACPS-PS (dynamic)	x	x	x
ACPS-PC (dynamic)	x	x	x
ACPS-PD (static)			x
ACPS-I (static)			x

Therefore, in operational context, the services provided by the ACPS-I are identical with those provided by its CC, the HC and PC being irrelevant (Table 1). Thus, the interaction between the ACPS-I with the (Fig. 5):

- ACPS-PD will dynamically identify the feasible abstract services that are available at the ACPS-PS level when the product instance is physically located in a certain place in the plant layout (*"context-specific product manufacturing knowledge"*);

- ACPS-PS will monitor in real time the status of the services provided by each production resource in terms of their availability and quality (*"production environment knowledge"*);
- ACPS-PC will assess the services in terms of availability and energy use (*"context execution knowledge"*).

4.3 ACPS-PD

Given the current state of a product instance (as received from the ACPS-PC), the ACPS-PD is responsible to provide the next abstract operation to get manufactured according to the customer requirements. The manufacturing process is described in an abstract form and is modelled as an OWL-S CompositeProcess (i.e. hierarchical task network). The decomposition of abstract processes until the level of executable web services is codified in an ontology (i.e. "function ontology") that captures the functional aspects (i.e. type of operation performed) of the available services at the plant level (Loskyll et al., 2012). To increase availability, the abstract process description is directly stored on a RFID tag attached to the product instance. Note that this ontology is describing the full functional capabilities provided by the ACPS-PS and not their feasible use in respect to the plant layout, the states of the ACPS-PS which the ACPS-I is responsible for, or intra-logistic operations (i.e. storage etc).

Consequently the abstract manufacturing process for a product type is defined by the HC (i.e. product design expert) of the ACPS-PD in an OWL-S tool, to define it in terms of abstract operations provided by the ACPS-PS components.. In our case the product design phase is assumed not to be a continuous process that needs prototyping or intermediate embodiment designs. Therefore the ACPS-PD is restricted to its CC component responsible to manage a simple hierarchical task network data structure for the abstract manufacturing process (Table 1). Anyway, it requires the interaction with the (Fig. 5):

- ACPS-PS to consider the functional capabilities of the production resources when constructing the abstract manufacturing process ("resource-specific product manufacturing knowledge");
- ACPS-I to filter the above hypothetical services provided by the ACPS-PS to those that are feasible reachable given the physical location of the product instance in the plant layout ("context-specific product manufacturing knowledge");
- ACPS-PC to provide the list with the hypothetically executable services (on a functional level) for the next abstract processing operation given the current state of the product instance ("product manufacturing knowledge").

4.4 ACPS-PC

ACPS-PC is basically responsible to perform correctly in respect to customer requirements an assigned product instance. As mentioned, in our demonstrator these requirements are provided from a control panel and includes preferences for both the desired end-product (e.g. name, e-mail, etc. which will be engraved on the key-finder) and/or the manufacturing process (i.e. the shortest delivery time, cheapest production costs in terms of energy consumption, etc.). Consequently, the ACPS-PC needs to consider all its integral components (i.e. HC, PC, and CC).

The ACPS-PC task clearly requires a tight interaction to exploit the knowledge from all ACPS types and presumes the dynamic orchestration of the provided services. Firstly, given the current state of the product instance, it requires "product manufacturing knowledge" from the ACPS-PD to get a list with the hypothetically executable services (on a functional level) for the next abstract processing operation. These services are further filtered by interacting with the ACPS-I ("context execution knowledge"), in terms of their feasibility, availability (e.g. running, available or malfunctioning) and the assessment of the energy consumption (e.g. low, medium, high). The data collected from the ACPS-I are further used to rank the services against different evaluation criteria (e.g. provided operation, equipment category, consumed resource, quality). After the best matching service is selected the corresponding ACPS-PS component is finally triggered for execution according to the client's requests taking into account its current status ("process execution knowledge"). Note that the assessment of energy consumption may be implemented inside the ACPS-PS components as well. Due to the limited computational power of the microcontrollers it belongs to the ACPS-I which runs on server.

4.5 Synthetic overview

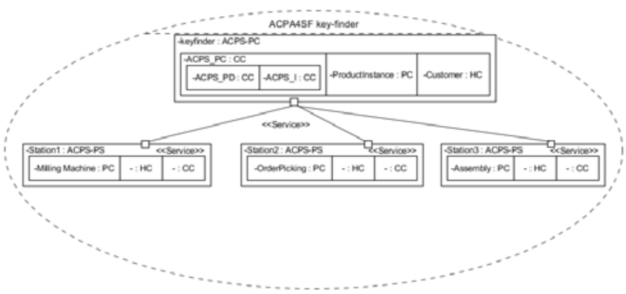

Fig. 6. The general structure of the SmartFactory^{KL} key-finder demonstrator

As may be observed in Fig. 6 the general structure of the key-finder demonstrator follows the classical hierarchical architectural control patterns (Diltis, Boyd and Whorms, 1991). This is due to the fact the ACPS-I and ACPS-PD does not need to play an active role, their degree of automation or delegation of control capabilities being a result of a pure cost-benefits analysis. Therefore the ACPS-I and ACPS-PD corresponding knowledge is considered to be static (i.e. basic assumptions) in the ACPS-PC component. Conversely, these two components, ACPS-I and ACPS-PD, would play an active role, if the infrastructure would control or supervise in real-time the layout of the production system in order to make feasible new manufacturing operations, or, in case of the product design, if it would control the very complex design space for a product based on the current state of the product instance and the reachable processing opportunities. Also note that if we consider the entire life-cycle of the production line with its continuous redesign, or whenever we have to consider the dynamics of an ACPS type, all the ACPS components are clearly involved in an active way.

5. DISCUSSION AND CONCLUSIONS

The paper presented an instantiation of our previous reference architecture where the humans will not just be its users, but elements of the system affecting its overall behaviour. Besides the classical control component, it integrates the physical and human ones as well. This is due to the real need to accommodate the on-going researches from the SmartFactory^{KL} facility (e.g. augmented reality, mobile interaction technology, virtual training of human operators) where the human is not merely playing a simple and clear role inside the control-loop, but is becoming a composite factor in a highly automated system ("man in the mesh"). Consequently, this requires solutions that support the humans to race together with the cyber-physical systems for the permanent development of a production system.

The reference architecture aims to serve as a guideline for on-going automation improvements, and consequently to accommodate all sorts of compromises in engineering the ACPA4SF along the gradual and long-term transition from a centralized to a distributed architecture. For its concrete instantiation a balance between the desirable short-term and long-term goals is required to deploy the necessary functionalities in a pure cost-benefits analysis.

6. ACKNOWLEDGEMENTS

This work is partially supported by the "CyProS" project, grant number 02PJ2477, funded by the German Government.

REFERENCES

Brynjolfsson, E. and McAfee, A. (2011). *Race Against the Machine*. Digital Frontier Press, Lexington.

Chituc, C.M. and Restivo, F.J. (2009). Challenges and Trends in Distributed Manufacturing Systems: Are wise engineering systems the ultimate answer? In *Second International Symposium on Engineering Systems*, MIT Press, Cambridge, Massachusetts, 1-15.

Diltis, D., Boyd, N. and Whorms, H. (1991). The evolution of control architectures for automated manufacturing systems. *Journal of Manufacturing Systems*, **10**(1), 63-79.

Gorecky, D., Garcia, R.C. and Meixner, G. (2012). Seamless Augmented Reality Support On The Shopfloor Based On Cyber-Physical-Systems. In: *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, San Francisco, USA, ACM.

Grubic, T. and Fan, I.-P. (2010). Supply chain ontology: Review, analysis and synthesis. *Computers in Industry*. **61**(8), 776-786.

Kasparov, G. (2010). *The Chess Master and the Computer*. Available at: http://www.nybooks.com/articles/archives/2010/feb/11/the-chess-master-and-the-computer/ [Accessed on 8.11.2013].

Lee, A.E. (2008). Cyber Physical Systems: Design Challenges. In International Symposium Object/ Component/ Service-Oriented Real-Time Distributed Computing, Orlando, USA, 363-369.

Licklider, J.C.R. (1960). Man-computer symbiosis, In *IRE Transactions on Human Factors in Electronics,* HFE-1, 4-11.

Loskyll, M., Schlick, J., Hodek, S., Ollinger, L., Gerber, T. and Pîrvu, B. (2011). Semantic Service Discovery and Orchestration for Manufacturing Processes. In *Proceedings of 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)*, Toulouse, France, 543-550.

Loskyll, M., Heck, I., Schlick, J., Schwarz, M. (2012). Context-based orchestration for control of resource-efficient manufacturing processes. *Future Internet*, **4**(3), 737-761.

McFarlane, D., Giannikas, V., Wong, A.C.Y., Harrison, M. (2013). Product intelligence in industrial control: Theory and practice. In *Annual Reviews in Control*, **37**(1), 69–88.

NIST (2013). *Foundations for Innovation in Cyber-Physical Systems*, Workshop Report, Available at: http://www.nist.gov/el/upload/CPS-WorkshopReport-1-30-13-Final.pdf [Accessed on 30.10.2013]

Pfeifer, R. and Bongard, J.C. (2007). *How the Body Shapes the Way We Think. A New View of Intelligence*, MIT Press.

Rajkumar, R. (2007). *CPS briefing*, Carnegie Mellon University.

Robu, V., Halpin, H. and Shepherd, H. (2009). Emergence of consensus and shared vocabularies in collaborative tagging systems, *ACM Transactions on the Web*, **3**(4), 1-34.

Schmitt, M., Meixner, G., Gorecky, D., Seissler, M. and Loskyll, M. (2013). Mobile Interaction Technologies in the Factory of the Future. In: *IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*. Las Vegas, Nevada, USA, 536-542.

Stork, A., Gorecky, D., Stahl, C., Loskyll, M., Michel, F., Sevilmis, N., and Weber, D. (2012). Enabling Virtual Assembly Training in and beyond the Automotive Industry. In: *The 18th International Conference on Virtual Systems and MultiMedia*, Milan, Italy, 347-352.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peetres, L. (1998). Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry, **37**(3), 255-274.

Weyns, D., Omicini, A. and Odell, J. (2007). Environment as a first class abstraction in multiagent systems. *Auton Agent Multi-Agent Syst*, **14**(5), 5–30.

Zamfirescu, C.B., Pirvu, B.C., Schlick, J. and Zühlke, D. (2013). Preliminary Insides for an Anthropocentric Cyber-physical Reference Architecture of the Smart Factory, *Studies in Informatics and Control*, **22**(3), 269-278.

Zhuge, H. (2010). Interactive semantics, *Artificial Intelligence*, 174(2), 190–204.

Zühlke, D. (2008). SmartFactory—From vision to reality in factory technologies. In *Proceeding of 17th IFAC World Congress*, South Korea, 82–89.