

# Comparative study of flexible and decentralized agent-based and service-oriented control architectures for production systems

Steffen Ruppert<sup>1</sup>, Frank Schönung<sup>2</sup>, Tanja Walz<sup>2</sup> and Julien Provost<sup>1</sup>

**Abstract**—The purpose of this paper is the presentation and comparison of control architectures for the modular factory. In the last few years, service-oriented architectures and agent systems evolved to the most suitable technologies for flexible and versatile production systems which cope with challenges resulting from shortened product life cycles and product customization. However, decentralization as a rising concept to reduce complexity in production systems is often ignored. Therefore, this paper introduces service-oriented and agent-based control architectures which are based on the *Production Priority Plan* guaranteeing more flexibility and enabling decentralized decision-making in order to react to unexpected events. The architectures are compared in test scenarios derived from future production system challenges. The study demonstrates the great potential of the flexible, versatile and decentralized architectural concepts for production systems by contrast with today's production controls. Furthermore, applications more suitable for a concept due to its specific characteristics are highlighted.

## I. INTRODUCTION

Present productions face various challenges and will be exposed to continuous change. Customers demand more individualized and specialized products leading to shortened product life-cycles, small lot sizes and a wide product variety. Therefore highly adaptable production systems are needed which offer a broad range of extensible functionalities and production configurations. Current production controls are not able to cope with the changes due to slow adjustment speeds and comprehensive modifications in the information infrastructure [1]. Furthermore, they are integrated with enterprise resource planning systems (ERPs) which are based on rigid work plans and centrally stored production data. In addition, digitalization enabled the entrance of cyber-physical production systems (CPPS), interconnected, communicating and intelligent machines, in productions to increase versatility. Legacy production controls consisting of numerous subsystems are incapable of exploiting the maximum potential of CPPS because of their fragmented composition. Besides, centralized information storage in ERPs limits the optimization possibilities in production control and impedes flexible counteraction on unexpected events [2].

Service-oriented architectures and agent systems (see II) are considered the most promising technologies for future production systems facing the aforementioned challenges. Therefore, this paper presents two flexible, versatile and

decentralized architectural concepts for production control: a service-oriented architecture and an agent-based architecture. Beyond that, the architectures are compared in test scenarios describing future challenges for production systems.

The remainder of this paper is organized as follows: in sec. II, applications of SOA and agent-based systems are presented and interpreted. The benchmark production system is explained in sec. III, before two architectural concepts for the modular factory are introduced in sec. IV. Sec. V compares and analyzes the concepts in scenarios derived from future production system challenges. The paper concludes with a summary of the presented results.

## II. BACKGROUND

This section presents use cases of service-oriented architectures and agent systems. Furthermore, the need for flexible, versatile and decentralized control architectures is derived. Conclusively, the principle of restful web services is introduced as preferred implementation approach for service-based architectures.

### A. Service-oriented architectures (SOA)

SOA is an architectural pattern whereby functionalities are modeled as services. The loosely coupled and independent composition in which services are easily added and published in registries is privileged for the use in flexible production systems. Due to the submission of calling conditions by service providers, clients execute required services by communicating with respective service providers [3].

The application of SOA in production systems is described by numerous researchers. [4] describes the great potential of SOA in production control. Here, a process-oriented planning concept is introduced enabling agility and pioneering to decentralization due to distributed sub-controls. In [5] SOA is applied in production control aiming at a higher flexibility and reconfigurability. Thereby, self-configuring process modules enable fast replacements to facilitate control.

### B. Agents

Agents describe definable units (hard- and/or software) with specific targets which strive to succeed by interacting with the environment and other agents [6]. Their characteristics enable autonomous target-oriented behaviors suitable for various flexible production scenarios.

The great potential of agents in production is described by [7] who propose dynamic control approaches and *plug & produce* concepts. Thus, flexible reconfigurations of manufacturing systems and improved production control monitoring are enabled by autonomous, cooperating agents. [8]

<sup>1</sup>Steffen Ruppert and Julien Provost are with the Technical University of Munich, Garching, Germany provost@ses.mw.tum.de

<sup>2</sup>Frank Schönung and Tanja Walz are with SEW Eurodrive GmbH & Co.KG, Bruchsal, Germany frank.schoenung@sew-eurodrive.de

reviews multiple studies on agent systems reflecting upon encapsulation, coordination, implementation, and acceptance. In conclusion, agent systems meet the requirements of agile, reconfigurable, distributed and intelligent production control systems. However, further development of agents is only successful in combination with other technology [9].

Aforementioned fundamentals and use cases confirm the great potential of service orientation and agents for production control. Nevertheless, flexibility, versatility, and decentralization are not satisfactorily implemented to enable a broad application. Hence, this paper introduces web service based control architectures supporting these characteristics: Flexible and versatile architectures are defined in [10] with “the ten types of manufacturing system flexibilities”. Decentralization by this means aims at the production control decision-making. Production control is defined by decentralized, distributed and autonomous decision-making entities considering current conditions of the production system.

### C. Restful webservices

Restful webservices are an architectural pattern for machine-to-machine communication which models entities as resources. Thereby, every entity is determined by a *Unified Resource Identifier* (URI) which provides access for adjustments and executions [11]. HTTP methods are utilized to transmit required information and to facilitate the scalability of web services. Architectural entities are classified into four categories: *Models* summarize physical and non-physical elements. Physical elements offer factual functionalities, whereas non-physical elements are responsible for internal processes. *Services* are strongly dependent on model elements since they describe their functionalities. *Registries* ensure the dynamic discovery of services to process clients’ requests. *Resources* function as interface. They enable bi-directional communication handling via requests and responses between server and client.

### III. BENCHMARK PRODUCTION SYSTEM

The presented control architectures are deployed in research of gear motor manufacturing. Gear motors are highly individualized products tailored for specific applications in various industries. Due to dynamic and competitive market conditions, gear motor productions face challenges as presented in sec. I. Subsequently, the production process is explained (see fig. 1):

Gear motors of a similar size are produced in small factory units (SFU) [12]. An SFU is an essential component in a modular production which aggregates specific production systems to manufacture a product or product family. It covers the entire production process which consists of both automated and manual operations. First, the gear is assembled in production islands. While passing several workstations gearwheels, bearings, shafts, and other components are combined in the housing. Thereby, the assembly order depends on product characteristics such as gear ratio and installation type. Following, the gear is filled with oil by automated oil filling machines. At the next production section, the engine is

mounted. Afterwards, the gear motor is checked at the quality control section via test procedures. Finally, it is painted and prepared for shipment.

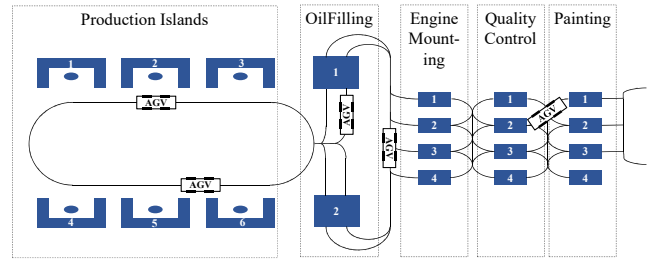


Fig. 1. Benchmark production system: Gear motor production in a SFU

### IV. CONTROL ARCHITECTURE DESIGNS

Prior to the presentation of the developed decentralized control architectures for the modular factory in IV-B and IV-C the *production priority plan* (PPP) is introduced as architecture underlying concept.

#### A. The Production Priority Plan (PPP)

The PPP is key to the flexibility of the control architectures. It is defined as follows:

**Definition:** The *Production Priority Plan* represents the relations of all production steps for a specific product with respect to their causal priorities in a network structure.

Fig. 2 shows a PPP consisting of 7 production steps. Compared to a work plan in which production steps are executed one after another, the PPP enables causally determined decision-making about subsequent production processes. For instance, since production step 1 is executed the production control may decide between the production steps 2, 3 and 4. As production step 5 requires the execution of the production steps 2, 3 and 4, the options for production control diminish as the decision-making proceeds. Here, intelligent high-level optimization planning is needed to avoid bottlenecks due to the reduction of options.

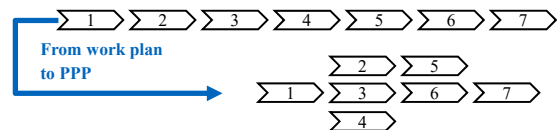


Fig. 2. The *Production Priority Plan* regained from a work plan

In conclusion, the PPP is a work plan break-up to regain the flexibility of rigid production procedures. This supports decentralized decision-making in production controls by enabling various proceeding possibilities. The PPP paves the way for comprehensive optimization methods taking operational key performance indicators into account (e.g. lead time, capacity utilization etc.).

Both following architectural concepts entail the PPP in order to feature decentralized and flexible characteristics.

### B. The service-oriented control architecture (SOCA)

The service-oriented control architecture (SOCA) was built following the development procedure described in [3]. Thereby, business processes, as well as existing production applications, are considered; which opens new ways for solutions and enables prompt functioning services.

In this section, the composition of the SOCA and its order workflow is introduced in IV-B.1 and IV-B.2.

1) *SOCA setup*: In the SOCA setup, functionalities are modeled as services which are listed in registries. The architecture constitutes a hierarchical structure (see fig. 3). Business processes (BPs) are embedded at the top of the architecture as order management leading entities. They are intelligent, with data equipped services which represent the customer orders. BPs are defined as

$$BP = BP(id, producttype, data) \quad (1)$$

with the order identification number  $id$ , the ordered product  $producttype$ , and all required production parameters  $data$ . Production parameters are derived from product features.

One level below SFU services (SFUSs) are applied which represent the functionalities of SFUs. Consequently, a SFUS is defined as

$$SFUS = SFUS(id, producttypes, data, ValueCreationUnits) \quad (2)$$

with the SFU identification  $id$ , all product types manufactured in the specific SFU  $producttypes$ , and the production parameters  $data$  needed to execute the manufacturing process.  $ValueCreationUnits$  represent the components of a SFU (e.g. assembly, quality control etc.) which are implemented according to the PPP. The SFUS are published in registries called by the BPs to perform a customer order. The next level encloses production services (PSs) which represent the functionalities of manufacturing modules. Manufacturing modules are product type specific entities to perform tasks corresponding to  $ValueCreationUnits$  (e.g. assembly for product x). Hence, PS are specified as

$$PS = PS(id, producttype, data, ValueCreationSteps, area) \quad (3)$$

with the identification number  $id$ , the manufactured product  $producttype$ , all production parameters  $data$  required to perform the functionalities, and the location  $area$  of the respective manufacturing module.  $ValueCreationSteps$  represent workstations within manufacturing modules which are executed according to the PPP. PSs are registered in registries which are called by SFUS when executing  $ValueCreationUnits$ .

While executing a  $ValueCreationStep$  services of production resources (e.g. machines), termed as *Processes* (Ps), are called and their functionalities are utilized. Again, the sequence is based on the PPP. Ps are defined as

$$P = P(id, data, position) \quad (4)$$

with the stage in the PPP  $id$ , the production parameters  $data$  needed to perform the functionality, and the production resource's location  $position$ . Ps are published in registries according to their  $id$  and  $position$ . They constitute the interface between the architecture and physical applications.

2) *Order workflow in the SOCA*: The order workflow starts with a BP created on a customer order containing the information utilized to manufacture the respective product. The BP calls a registry in which SFUSs are published. The registry returns a number of SFUSs which are capable of manufacturing the ordered product and their respective calling conditions. The BP chooses and calls the most suitable SFUS by transmitting the required information.

The SFUS performs its  $ValueCreationUnits$  according to the PPP. Thereby, registries are enquired which incorporate PSs and their calling conditions. SFUSs evaluate registry responses and call optimal PSs by transferring required production parameters.

A PS executes its  $ValueCreationSteps$  based on the PPP. It requests Ps in registries which return solvent and competent production resources. The PS determines which P to execute based on the current production system's condition. This is deployed by consigning production parameters via requests. Decentralized decision-making once a registry response is evaluated guarantees that optimal services are executed with respect to the production system's conditions and objectives (e.g. capacity utilization, lead time, shortages). Services of a hierarchically higher level take the decision-making authority on lower level services. Hence, the decisions are decentralized in a temporal and architectural manner.

### C. The agent-based control architecture (ABCA)

The agent-based control architecture (ABCA) represents an architecture in which agents utilize services offered by production resources to achieve their target: the manufacturing of a product.

The section presents the composition of the ABCA in IV-C.1 and describes its order workflow in IV-C.2.

1) *ABCA setup*: In the ABCA, customer orders represent smart agents (SAs) equipped with the PPP and respective production parameters. They exert services of production resources directly which results in a flat hierarchy (see fig. 4) A SA is defined as

$$SA = SA(id, producttype, position, data, processes) \quad (5)$$

with the identification number  $id$ , the ordered product  $producttype$ , the product's location  $position$ , the required product parameters  $data$ , and the production steps  $processes$  according to the PPP.

A *process* (P) models a production resource's functionality as service. They are specified as

$$P = P(id, data, position) \quad (6)$$

with the stage  $id$  in the PPP, the production parameters  $data$  to perform the functionality, and the production resource's location  $position$ . Ps are offered in the *production network*

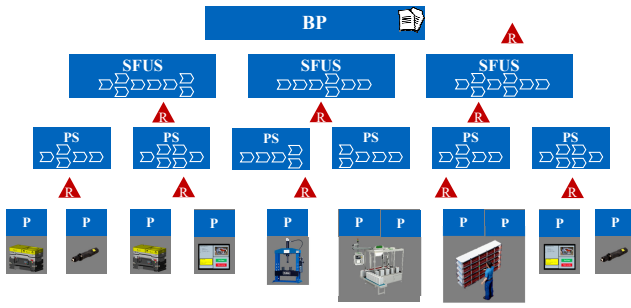


Fig. 3. The SOCA hierarchical structure

by their respective production resource and represent the interface between the architecture and physical applications.

2) *Order workflow in the ABCA*: As a prerequisite for the ABCA order workflow, SAs contain the PPP based on Ps and the entire product parameters necessary for manufacturing. Consequently, they are extensive to build and potent entities. SAs navigate through production independently. According to the PPP they request Ps by broadcasting over all production resources. Available and solvent production resources respond with their calling conditions. The responses are evaluated by the inquiring SA considering the PPP and production system's conditions (e.g. capacity utilization, lead time etc.). Subsequently, the optimal production resource is directly called to execute the requested service.

Note that the SA's decision-making is based on currently received responses and present production system's conditions. As a result, the ABCA is indicated as decentralized in both, a temporal and architectural manner.

#### D. Architectures' implementation

For the architectures' implementation restful webservices are applied in Java Eclipse. The aforementioned structures are employed by assigning entities to the four categories and in which every entity is addressable by its unique URI. Fig. 5 and 6 present the implementations of the architectures.

In both concepts model elements are defined by their characteristics. Inheritances describe relations among elements featuring similar characteristics. Service elements constitute the functionalities of respective model elements. The entirety of services is structured according to the model elements what underlines the strong dependency. For the SOCA, registry elements support its hierarchical structure and guarantee successful order workflows. They incorporate published services and associated model elements to process incoming requests from clients by service discovery. Just one registry element exists in the ABCA, the *production network*. Services are offered by respective model elements within the network to process client requests. The *production network* serves as a key element for communication in the ABCA. In both concepts communication proceeds via resource elements. The SOCA contains one for every entity type, whereas in the ABCA resource elements exist for different model elements and the *production network*.

Clients in the SOCA are alternating. Services of higher hierarchical level function as clients for applications of a

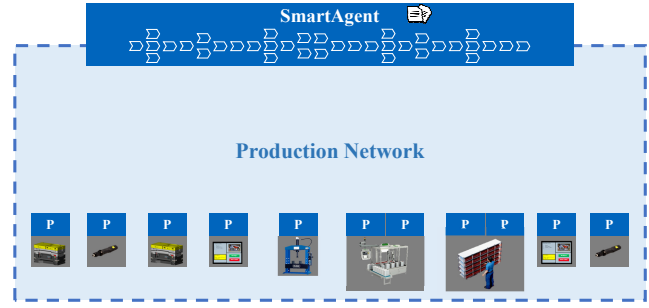


Fig. 4. The ABCA flat structure

lower hierarchical level. In the ABCA, exclusively, SAs are clients for all applications.

#### V. TEST SCENARIO BEHAVIORS AND COMPARISONS

This section focuses on the comparison of the presented control architectures (see sec. IV) with each other and classic production control architectures (CPCA - see box in V) in test scenarios. The test scenarios focus on future architectural and communication challenges faced by productions in the course of a high product variety, customized products and advancing digitalization [6], [13], [14].

##### *Classic production control architectures (CPCA):*

In CPCA an ERP spans over the architecture and processes customer orders to work orders. Thereby, part lists and work plans are utilized. For operation, work orders are transferred to manufacturing execution systems (MESs) which perform production and capacity planings. As a result, work orders and their production steps are assigned to specific manufacturing facilities and production resources. The ensuing order management is characterized by a distributed composition consisting of a host computer and legacy systems which results in fragmented and coupled system infrastructures. These systems constitute the interface to physical applications. They are supplied with production parameters or product data by either ERPs or MESs. Hence, consistent communication is essential for successful order management. Furthermore, the host computer and legacy systems are highly customized for specific responsibilities; which impedes modifications [2].

#### A. Test scenarios

The following six test scenarios are considered in comparisons of the control architectures (see tbl. I):

1) *Adding a new production resource (ANP)*: Due to further product customization and the entry of CPPS into shop floors, productions and their architectures need to cope with the implementation of new production resources. Thereby, production capabilities and capacities are extended by new functionalities and additional utilization possibilities.

2) *Introducing a new product (INP)*: Shortened product life cycles and high innovation power lead to more new

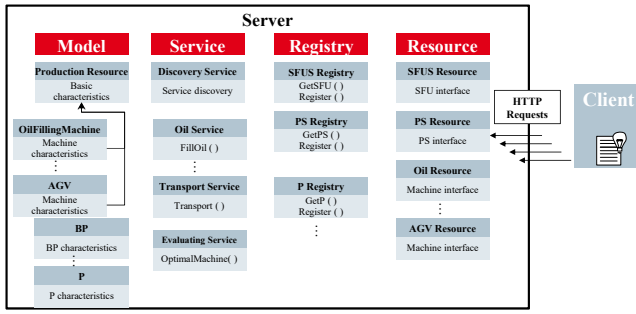


Fig. 5. The SOCA implementation via restful webservice

products in less time. Production systems and their architectures need to integrate various processes and adapt to heterogeneous products.

3) *Handling a wide product variety (HPV)*: Individualized products and the objective of implementing *one-piece-flows* in productions result in a high product variety based on different production parameters and steps. Production systems need to be versatile and adaptable to enable a wide product variety.

4) *Changing the production layout (CPL)*: Versatile and adaptable production systems transform continuously. Layout modifications become routine processes which are expected to be easily realized in the architecture.

5) *Failure of production resources (FPR)*: Failures of production resources lead to delays, production downtimes, and profit losses. Control architectures need efficient mechanisms to bypass failed production resources.

6) *Network communication problems (NCP)*: This scenario depicts the increasing dependency between production systems and centralized ERPs and analyzes the effect of communication problems between the high-level planning tools and production controls on the manufacturing system.

## B. Comparisons

This paragraph presents the detailed comparison of the control architectures in the most significant scenarios regarding their characteristics: *ANP*, *HPV*, and *NCP*. It considers the architectural composition, functioning, and implementation.

1) *ANP*: Introducing new production resources and implementing their functionalities in *CPCA* is extensive since multiple adjustments in various systems may be necessary: New production resources and their functionalities are created in the ERP to enable the integration in existing work plans. Consequently, the production parameters are adapted. New production resources and functionalities also lead to changes in the host computer and legacy systems for production control. Following these changes, the interoperability of the systems needs to be guaranteed.

In this scenario, the **SOCA** is adjusted on every hierarchical level: The new functionalities are offered as services. Thereby, the services are published in registries with specific production parameters. Then PSs are extended or newly established with adjusted *ValueCreationSteps* and *data*. Registries, listing affected PSs, are modified as well as SFUSs requesting these registries. Their

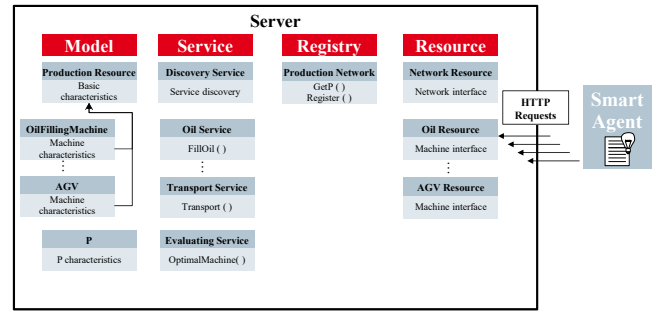


Fig. 6. The ABCA implementation via restful webservice

*ValueCreationUnits* and *data* are enhanced by modified PSs. Finally, the production parameters of the BPs are customized.

For the **ABCA**, this scenario requires only a few architectural changes due to its flat hierarchy: New functionalities are modeled as Ps in the architecture and linked to the new production resource. For utilization, SAs are equipped with information on the new Ps. Ps are considered in the PPP and *data* is extended by required production parameters.

In summary, adding a new production resource is best implemented with the **ABCA**. Due to its flat hierarchy and the adjustments in SAs, the architecture is advantageous to the **SOCA** and **CPCA** in which comprehensive architectural modifications are needed.

2) *HPV*: A wide product variety in **CPCA** affects the work plans in the ERP. Each variant is described by its tailored work plan. Thereby, parts lists and production steps are assigned. Furthermore, the host computer and legacy systems require modifications to handle the product variants effectively. Due to system-wide adjustments for each variant, handling a wide product variety in **CPCA** is complex.

In the **SOCA**, customized services according to variants are implemented in the hierarchy. New *ValueCreationSteps* and *ValueCreationUnits* are developed considering the variants' PPPs. Hence, BPs only contain required product parameters to manufacture new product variants. Due to the independently selectable granularity and scalability of services, entities with a high reusability are built covering all variants. The more variants are implemented in the **SOCA**, the better new variants are executable.

In the **ABCA**, a new SA is created for every variant. Thus, a new PPP is built for each variant. Prior to execution, the SA is equipped with the PPP and the required production parameters. A large number of variants results in extensive SA development with a unique SA for each variant. Conclusively, the **SOCA** handles a wide product variety best which results in a high service reusability. The reusability significantly reduces the complexity for executing diverse product variants, compared to the **ABCA** and **CPCA** which require comprehensive modifications for each variant.

3) *NCP*: The **CPCA** are strongly depending on the ERP because of its centralized information storage and the consequential information spread. Network communication problems block the information flow to the host computer and legacy systems. In consequence, their functionalities



are not applied and the production is not executed leading to downtimes. Due to its coupled and fragmented systems depending on the ERP, CPCA does not feature robustness. This scenario does not affect the SOCA, since the production control is decentralized and independent of ERP data once a BP is launched. The entire information for executing BPs lies in the SOCA. The PPP is implemented by services, the production parameters are stored in BPs and transferred to respective entities. As a result, the SOCA continues notwithstanding network communication problems occur which underlines its robustness.

Also, the ABCA is not affected by this scenario due to its decentralized decision-making and information storing in SAs. SAs independently control the manufacturing by applying services of production resources according to the PPP. Hence, the manufacturing is executable confirming ABCA's robust structure.

Subsequently, the two presented control architectures cope with network communication problems because of their decentralized information storage and independent decision-making. This enables autonomous production controls and increases the architectures' robustness.

### C. Discussion

Finally, the evaluation of the control architectures in test scenarios (see tbl. I) states that the SOCA and the ABCA manage future challenges of production systems better than CPCA. Most significant reasons are:

- CPCA feature a fragmented and coupled set-up which complicates architectural changes due to many modifications in systems and interfaces.
- The SOCA and the ABCA support a high flexibility in production control because of the PPP based production procedure.
- The SOCA and the ABCA increase the production system's versatility by easy implementing services, guaranteed web service based interoperability and clear architectural structures.

|      | ANP | INP | HPV | CPL | FPR | NCP |
|------|-----|-----|-----|-----|-----|-----|
| CPCA | -   | -   | -   | -   | -   | -   |
| SOCA | -   | =   | +   | -   | +   | +   |
| ABCA | +   | +   | -   | +   | -   | +   |

TABLE I

SUMMARIZED QUALITATIVE EVALUATION IN TEST SCENARIOS

Furthermore, the study identifies specific characteristics of the architectures. Due to a high service reusability, the SOCA suits production applications with a high product variety whereby the scalability and granularity of the architecture are defined individually. The architecture spanning character of the BPs supports an easy information exchange in case of order modifications. Apart from that, the ABCA is beneficial to production applications with heterogeneous products which require diverse PPPs, each implemented in a unique SA. Architectural changes are simple to carry out due to its flat hierarchy. Hence, the decision which concept to apply

depends on the challenges that the specific production system will most likely face.

## VI. CONCLUDING REMARKS

This paper introduces two flexible, versatile and decentralized control architectures: the SOCA and the ABCA. They are compared with classic production controls in scenarios describing future production system challenges. The novel underlying concept is the PPP ensuring flexibility and withal enabling decentralized decision-making. Thereby, the control architectures are empowered to optimize themselves and to react effectively to unexpected events. The test scenarios showed that the implementation of the control architectures via restful web services supports flexibility in production control and versatility of the production system. The investigations highlight that the developed concepts cope with future challenges better than classic production controls due to their extensible setup and decentralized decision-making according to current events. Furthermore, the results emphasize that each concept suits specific applications best. Hence, the control architectures enable further evolution and optimization of production systems in the environment of digitalization and customization based on their characteristics.

## REFERENCES

- [1] M. Waibel, L. Steenkamp, N. Moloko, and G. Oosthuizen, "Investigating the effects of smart production systems on sustainability elements," *Procedia Manufacturing*, vol. 8, pp. 731–737, 2017.
- [2] M. Foehr, J. Vollmar, A. Calà, P. Leitão, S. Karnouskos, and A. W. Colombo, "Engineering of next generation cyber-physical automation system architectures," in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*. Springer, 2017, pp. 185–206.
- [3] T. Erl, *SOA design patterns*. Pearson Education, 2008.
- [4] L. Ollinger, J. Schlick, and S. Hodek, "Leveraging the agility of manufacturing chains by combining process-oriented production planning and service-oriented manufacturing automation," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 5231–5236, 2011.
- [5] S. Scheifele, J. Friedrich, A. Lechler, and A. Verl, "Flexible, self-configuring control system for a modular production system," *Procedia Technology*, vol. 15, pp. 398–405, 2014.
- [6] P. Leitao, 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, 2016.
- [7] A. Rocha, G. Di Orio, J. Barata, N. Antzoulatos, E. Castro, D. Scrimieri, S. Ratchev, and L. Ribeiro, "An agent based framework to support plug and produce," in *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. IEEE, 2014, pp. 504–510.
- [8] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979–991, 2009.
- [9] A. W. Colombo, S. Karnouskos, J. M. Mendes, and P. Leitão, "Industrial agents in the era of service-oriented architectures and cloud-based industrial infrastructures," in *Industrial Agents*. Elsevier, 2015, pp. 67–87.
- [10] H. A. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *International journal of flexible manufacturing systems*, vol. 17, no. 4, pp. 261–276, 2005.
- [11] E. A. Marks and M. Bell, *Service Oriented Architecture (SOA): a planning and implementation guide for business and technology*. John Wiley & Sons, 2008.
- [12] J. Soder, *Use Case Production: Von CIM über Lean Production zu Industrie 4.0*. Springer, 2014, pp. 85–102.
- [13] M. M. Tseng and S. J. Hu, "Mass customization," in *CIRP Encyclopedia of Production Engineering*. Springer, 2014, pp. 836–843.
- [14] G. Morel, P. Valckenaers, J.-M. Faure, C. E. Pereira, and C. Diedrich, "Manufacturing plant control challenges and issues," *Control Engineering Practice*, vol. 15, no. 11, pp. 1321–1331, 2007.