Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 81 (2019) 541–546

www.elsevier.com/locate/procedia

52nd CIRP Conference on Manufacturing Systems

Platform-based service composition for manufacturing: A conceptualization

Jonathan Fuchs[a,]*, Sascha Julian Oks[b], Jörg Franke[a]

[a]Institute for Factory Automation and Production Systems, Friedrich-Alexander University Erlangen-Nuremberg, Egerlandstraße 7-9, Erlangen 91058, Germany
[b]Chair of Information Systems, Innovation and Value Creation, Friedrich-Alexander University Erlangen-Nuremberg, Lange Gasse 20, Nuremberg 90403, Germany

* Corresponding author. Tel.: +49-9131-8527967; fax: +49-9131-302528. *E-mail address:* jonathan.fuchs@faps.fau.de

Abstract

With the implementation of cyber-physical production systems, a multitude of isolated industrial services utilizing versatile hardware and software components in heterogeneous and rigid system architectures are an obstacle for the consequent extension of holistic automation within production networks. To address this issue, we conceptualize a platform-based system architecture enabling value-adding service bundles based on and extending proven architectural styles. Our proposal also aims on qualifying communication systems to work transboundary in value-adding networks. The presented architecture is applied to a use case in context of maintenance. Our findings provide an adaptive conceptualization for application-specific service composition and integration.

Keywords: IIoTS platform; service compostion;

1. Introduction

With the introduction of cyber-physical production systems (CPPS), novel configurations of information and communication technologies (ICT) are shaping the industrial production environment [1].

The extending adoption of ICT in modern production processes allows far-reaching digitalization of industrial value creation. In particular, increasing hardware performance combined with miniaturization and constantly improving resource efficiency enables the use of smart components up to the level of embedded systems. Frameworks like the *reference architecture model Industry 4.0* or the *asset administration shell*, are intended to align the development of Industry 4.0-capable solutions and thus, to further automate and optimize both production processes and especially cooperation between organizations in value-adding networks. [2, 3]

In this context, industrial, software-based services have become more fundamental: Products, services and processes are meaningfully combined into so-called *smart services*, enabling companies to establish new business models and revenue architectures [4]. However, such services are often not designed for fully integrated utilization and thus organizations face severe challenges transferring them into value-adding profit. This is attributable to two aspects: First, existing services are often designed with the logic of proprietaries due to a lack of standardization and mostly intended for a static, previously defined purpose. Second, the coupling of individual atomic entities to value-adding service bundles is not an aspect that is considered during development. This results in barriers to fully automated and smart interactions between entities of industrial service ecosystems. However, these are prerequisites for service systems engineering, which enables the systematic exploitation of the potentials of digitalization [5, 6]. In this context, the availability of automated service composition is an essential success factor, especially for increasing productivity and the further development of existing, but also for the design of entirely new business models. For this reason, the automated

10.1016/j.procir.2019.03.152

composition of services within the scope of platform ecosystems is a very relevant topic from both a technical and a business perspective.

This paper is organized as follows: Section 2 defines the term *industrial digital service*, delimits the perspective relevant to this work, briefly introduces relevant communication concepts (2.1) and distinguishes between service orchestration and service choreography (2.2) in the context of industrial information technology (IT) platform concepts (2.3). In addition, the obstacles, e.g. caused by heterogeneity in manufacturing systems are classified (2.4). Section 3 deals with the proposal of an applicable platform-based system architecture enabling value-adding service bundles. Section 4 summarizes the proposed architecture in a nutshell and gives an outlook on further steps for subsequent development and practice-oriented implementation of the concept.

2. Digital industrial service systems

Industrial digital service systems generally describe service systems based on the use of ICT [7]. However, in academic discourse as well as in the field of applied research and development, there are several understandings of the term *service* in industrial environments [8]. In context of the proposed architecture, we do not limit our view to certain areas of manufacturing, e.g. the shop floor or high-level corporate functions, as they are structured in the multi-level model also known as automation pyramid [9]. Following [10] we understand the described system as a nonhierarchical network, in which ubiquitous communication is feasible through suitable infrastructures and propose system-relevant additions.

From an organizational viewpoint, industrial services can be seen as direct or indirect sources of revenue for providers [11, 12] and are typically divided into groups [13]: Following this segmentation, particularly the two categories *maintenance services* and *operational services* comprise elements that are relevant in context of this paper.

2.1. Communication concepts for industrial service networks

Digital transformation has led to the development of diverse concepts and solutions for the use, integration and composition of Internet-based services as well as digital business models. However, these concepts often cannot be transferred easily to systems of manufacturing companies and their partner networks without adaptations of the communication technology requirements [14]. For the implementation of service-oriented architectures (SOA) in industrial environments, the concept *Enterprise Service Bus* (ESB) has been established with regard to communication and offers an important basis for future development of communication solutions that enable generation and barrier-free use of value-adding service bundles. A performant example is the *Virtual Automation Bus* (VAB), which was developed within the framework of the Industry 4.0 middleware *Basys4.0* and enables end-to-end communication between various production participants on basis of the *Asset Administration Shell* (AAS) [15]. The *NAMUR Open Architecture* (NOA) pursues a different approach: This process industry standard supplements the classic automation pyramid with open interfaces, e.g. on basis of industrial communication protocols like *Open Platform Communications Unified Architecture* (OPC UA) but does not strive for the conversion of such systems to non-hierarchical service networks [16]. *Open Process Automation* (OPA) represents a contrary approach: In future control systems will be completely redundant and system functions will be provided platform-based according to this approach [17].

2.2. Industrial services and service composition

Especially in non-industry-specific application areas, i.e. in the business-to-customer (B2C) segment, it is evident that digital platforms and services provided on them enable successful business models. However, such models are not easy to transfer to industrial environments, since B2C applications are mostly characterized by a multitude of one-time transactions between the participants and not by long-term relationships [18].

According to the paradigm of service-oriented computing (SOC), services are used as encapsulated basic elements to map complex business logics. Services are abstract software elements or interfaces that provide other applications with standardized access to application functionality via a network [19]. In order to meet the requirements formulated by Papazoglou [20], services must fulfill three essential characteristics: (1) Technological independence, (2) location transparency and (3) loose coupling.

Based on the SOC concept, existing services can be combined in bundles. In this context, a distinction is made between service orchestration and service choreography. Orchestration describes a flexible combination of internal or external services into a composition that defines a certain business process. In contrast, services in a choreography describe only their respective purpose without any larger context. The focus is therefore on service functionality and the interaction between individual processes. [21]

2.3. Service-oriented architectures, micro services and platform-based service provision

A SOA is a "[…] multi-layered, distributed architectural principle of an ICT-based system. Parts of applications, for simplified process integration, are encapsulated as business-oriented services and certain design principles are taken into account" [22]. The introduction of SOA is intended to transform the structure of monolithic IT systems into a network of distributed services without a defined hierarchy of available components in these systems. In order to achieve this result, applications must be broken down into their logical components. Among others, a SOA-based system must fulfill the following non-functional properties: (1) Modular expandability without changing the inherent logic of the holistic system, (2) redundancy and resilience as well as (3) scalability.

Services are key elements in this context. They are used to map subfunctions of extensive applications. Especially the concept of *micro services* plays an important role. They are regarded as software components that can be operated without interdependencies. Each micro service has a specified functionality within a system and its communication format is decisive. The principle *smart endpoints - dumb pipes* reflects that approach: The infrastructure used for message transmission between components serves exclusively as a message router while programmatic intelligence is in the endpoints (the services). They receive requests, process the input data and produce a response. [23]

Accompanying, some of the most important foci of the fourth industrial revolution are primarily set on linking and interacting between machines, processes and personnel, as well as automating complex business processes through servitization [24]. These efforts in industrial production are subsumed under the term industrial internet of things and services (IIoTS). In this context, cloud-based platforms for the provision of data-driven smart services are also gaining importance. The integration performance expected from IIoTS platforms becomes clear in the sheer amount of the different user roles and corresponding hurdles [25].

2.4. Challenges in digitized production networks

Enterprises face far-reaching challenges in the preparation for the future of production, with strategic hurdles to be tackled on a broad basis. These include highly globalized markets, highly individual customer demands, extensive automation efforts and rapidly evolving technologies. In particular, the factors of automation and advancing technologies pose strong operational challenges causing profound changes in many business areas [26].

In context of Industry 4.0, platform solutions that make services available online play a major role, with many different providers creating heterogeneity and increasing complexity in the market. The various solutions are difficult to combine, and individual services cannot be integrated easily into other platforms. These hurdles prevent the emergence of efficient platform ecosystems. Cross-company communication is also difficult due to different limitations. They include security requirements, but also the concern of manufacturing companies to disclose internal data and thus the acceptance of competitive disadvantages.

It is also difficult to identify individual services, which should be meaningfully combined in service bundles. Apart from planning support tools such as [27], a platform environment must also support the user in mastering complexity and providing suggestions for possible service combinations. This is the basis for monitoring and evaluating service bundles in terms of their effectiveness and efficiency and to optimize combinations during use.

Solutions for the administration of service networks like *Consul* do not offer adequate functionalities to meet the technical requirements needed. Also, the sole use of an ESB is not sufficient to combine services and thus foster value creation and to maximize the degree of automation in trans-organizational networks.

3. A platform-based system architecture enabling value-adding service bundles

In addition to common semantic communication protocols and ESBs, service platforms utilizing these concepts are a promising approach to solve the challenges outlined in section 2.4. They allow to link atomic service components and benefit from pre-arranged service bundles of different providers both internal and beyond company boundaries. For implementation, the overall system must be modularized, and each component must be regarded as an independent service providing its own specified interfaces for interaction.

In the context of this paper, we assess industrial digital services as software-based business-to-business (B2B) solutions offered on digital platforms online. Furthermore, we differentiate between atomic, not further decomposable service entities and combined value-adding services, so-called *service bundles*, built with the help of atomic services. Such service bundles can be generated automatically or user-based, however users must be supported by the system when composing the context- and company-individual service bundles.

The proposed architecture is clustered in the context of *Platform as a Service* (PaaS) systems. This concept extends the functionality offered by the *Infrastructure as a Service* (IaaS) approach by integrating development and runtime environments. This enables the development and utilization of cross-platform application components. However, the focus of the developed system architecture is not exclusively on the provision of runtime environments for user-friendly service integration, but on the simplification and standardization of communication between individual service components within a platform and between platforms of various providers. In the context of our developments, we use a sustainable communication substructure that enables the implementation of platform-based service systems especially in manufacturing technology. We also follow the concepts that focus on the creation and use of non-hierarchical IT networks.

3.1. System architecture and components

Following, we propose a conception of a holistic system architecture in which all hard- and software-based elements from the shop floor to the ERP level as well as the user can be integrated and linked in terms of ICT. The different user roles within a company and its partner enterprise network are thus offered extended capabilities for controlling, evaluating and eventually increasing the productivity of their cooperation. In the developed architecture several relevant aspects are considered. Fig. 1 shows the different entities within the system: The shop floor level contains a variable number of different production facilities. These are connected to the platform via the *message service bus* (MSB) and thus also to the distributed services. The platform solution itself integrates different service types. These include components of the platform management, e.g. the service directory as well as

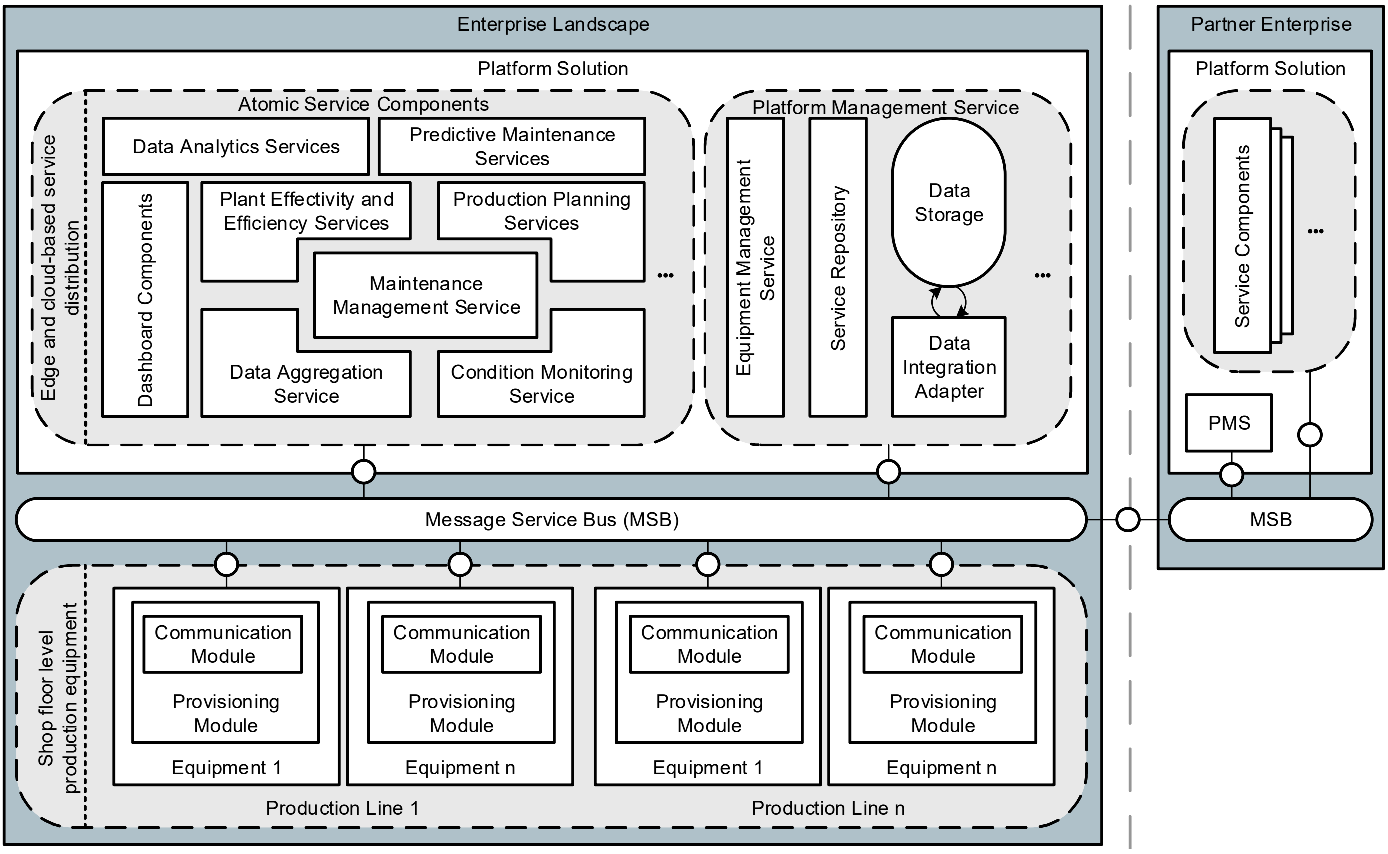

Fig. 1. Overview of the proposed cross-enterprise service-oriented platform architecture.

atomic service components from various providers, e.g. data analytics services or visualization components. In addition, the value-adding linkage based on services between partner companies is addressed. The following general explanation of the architecture is based on the *4+1 view model* by Kruchten, whereby the scenario is explained in chapter 3.2, as part of the application case-driven implementation. [28]

The **Logical View** illustrates the end-user functionality by describing the individual modules that the system provides to the user. The proposed architecture acts as a generic framework for the implementation, orchestration and especially choreography of different, predominantly software-based modules. For the implementation of this system distribution and configuration of functional units as well as their integration into the platform management are necessary. This allows simple networking of standardized entities in which different functionalities are encapsulated.

Non-functional elements are described in the **Process View**. These include interaction mechanisms, the performance and scalability of the architecture. Sophisticated systems can execute different operations asynchronously. This allows processes to run in parallel, new instances of subprocesses can be called and additional service modules may be integrated at runtime. The mapping of functional modules in independent, encapsulated micro services ensures component autonomy. In this way, different hardware versions can be used effectively at different architectural levels. In this context, we differentiate between the shop floor, edge, ERP and cloud levels. The limiting factor in this context is the communication structure, i.e. the maximum number of messages that can be exchanged simultaneously between the different services and the possible number of parallel connections. The use of a lightweight message bus and event-based information exchange ensures reliable and fast communication.

The **Physical View** focuses on mechanisms of software distribution on different hardware levels. We favor container solutions for the simplest possible distribution. The implementation of containerized software enables the fast and easy distribution of services from different providers, since most of all computer systems support runtime environments for solutions such as Docker, Rocket or LXD and thus software can also be used in heterogeneous systems independent of platform and location.

The **Development View** illustrates the organization and interaction of the individual modules in the overall system. This module-based view makes it easy for users to integrate specific individual services into extended service bundles. The abstract design of semantic interfaces simplifies the integration of new functions into the overall system. Each service must be developed in such a way that it describes its ability, necessary input for function fulfillment and expected output for other services in a comprehensible way. The role of the integrator is assumed by the platform management service (PMS).

3.2. Use case driven implementation

In the following, we give an example of a concrete solution to illustrate the advantages of service-based consideration and subdivision of an extensive business process enabled by our proposed architecture. The automated, service-based creation of an internal or external maintenance order for a facility that has failed in production serves as an implementation example. Fig. 2 shows the elements involved. The process flow can be seen in Figure 3. The focus is exclusively on the presentation of communication routes and consecutive steps, not on the definition of concrete data formats or exchange mechanisms and is designed as follows: The automation system is monitored by the condition monitoring service (CMS), deviations and error states are detected without delay, visualized via a dashboard component and immediately forwarded to the maintenance management service (MMS) in a semantically described manner. The core task of the MMS is the aggregation of additional information from various data sources that further describe the error which is uniquely identified by an ID. Such metadata can be maintenance guidelines or instructions, but also statistics on this specific error. The collected data then serves the respective decision maker, e.g. the maintenance manager, as an extended decision basis for the internal or external allocation of the maintenance order. If the plant can be serviced internally, the collected information is transferred to the production planning service (PPS). If maintenance is only viable by the equipment manufacturer, the data package is forwarded and processed there within the order management service. For the implementation we use the event-based messaging protocol WAMP. In addition to originally exchanged information, remote procedure calls (RPCs) can be triggered that allow loosely coupled services to initiate method calls at their communication partners. This way information can be requested and exchanged between the entities as required, fulfilling high industrial security requirements. The automatic creation of a service request enriched with important metadata in the equipment manufacturers system is done in a few seconds, depending on the approval response time of the maintenance manager.

Service location also plays an important role in the case described in this paper: Examples of services that are integrated in the edge, close to equipment, are the CMS or the data aggregation service (DAS), which is responsible for aggregating and filtering information from the shop floor and helps keeping the data volume to be transported low. The components of platform management and services like PPS or MMS, on the other hand, are located at the MES or ERP level, since there are fewer requirements regarding latencies or reliability. Elements such as data analytics or predictive maintenance are clearly to be found at the cloud level, as long-term evaluations based on extensive data records are the goal here. In such a distributed architecture, each service offers its own functionality and must be deliberately located in the overall system, whereby communication aspects play an important role for many applications. Latencies, failure probabilities and possible consequences for productivity are considered, making the location of components a central task.

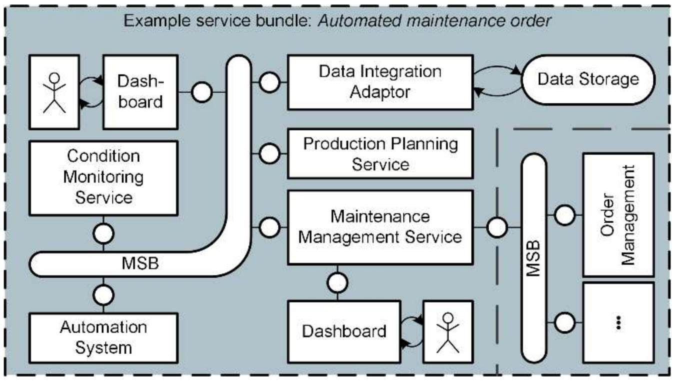

Fig. 2. Service entities composing the service bundle automated maintenance order.

3.3. Advantages concerning value creation

The proposed architecture for service composition addresses the realization of potentials associated with industrial digitalization. It fosters the establishment of platform ecosystems that are a prerequisite for shifting from static to dynamic, adaptive and context-dependent production and value creation processes. These can be used within a single company, but unfold their full potential especially by cooperative, trans-organizational value creation in ad hoc or strategic corporate networks. As shown by the examples of CMS and MMS, the effectiveness and efficiency of processes and thus the

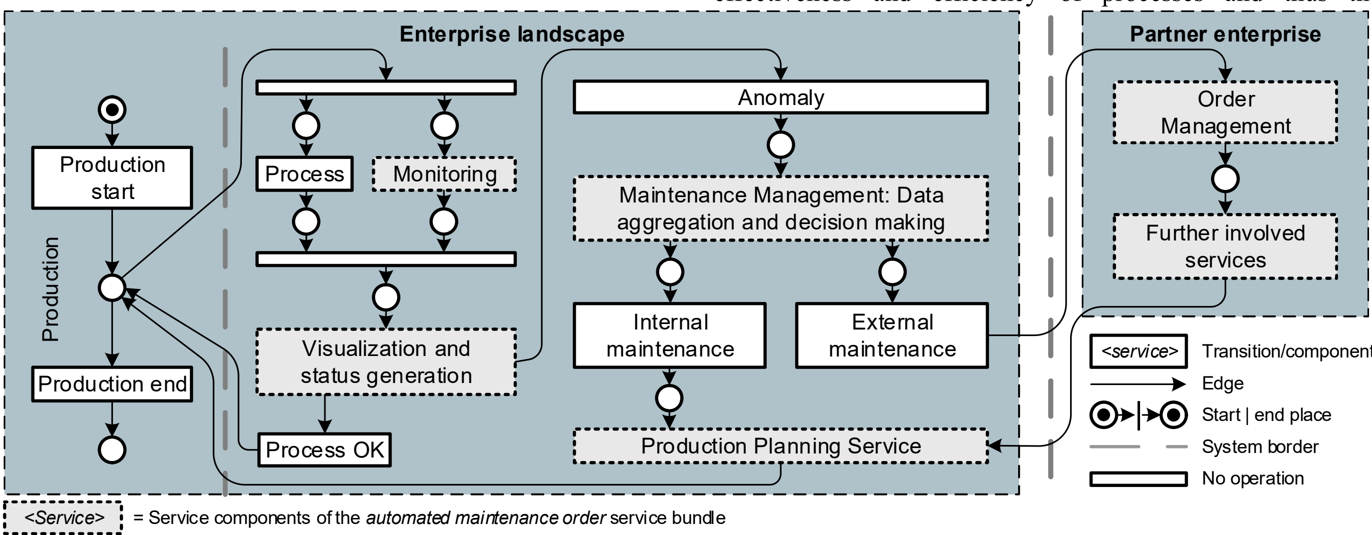

Fig. 3. Process view of the automated cross-company maintenance order service bundle.

productivity can be increased by modular service compositions. Furthermore, they have a significant influence on the design of business models based on these processes.

For these reasons, the proposed architecture and the service bundles generated in its context give a valuable contribution to address market-sided trends, such as highly individual customer demands, smart products and hybrid value creation. Due to standardized integration and easy combination of services of different providers in a platform solution, the obstacles heterogeneity represents to trans-organizational value creation are reduced.

4. Conclusion and Outlook

This research presents a holistic system architecture that allows software-based industrial services to be composed into value-adding service bundles. This is achieved by the systematic mapping of components through micro services and the use of a ubiquitous and lean message middleware. In addition, the cross-company coupling of services in value-adding networks is addressed. The chosen example clearly shows that fast and transparent provision of relevant data and information is an important support for decision makers. Based on the work described here, we will focus on the definition and development of further industry-relevant service bundles based on the method described in [27, 29] as well as on the integration and interaction between platform solutions of different providers. Examples of further service bundles are easy to identify and range from initial operation activities for plants to smart order management and quality optimization through customer feedback loops during product life cycle.

Acknowledgements

The research presented in this paper was funded by the German Federal Ministry of Education and Research in the course of the project PRODISYS (FKZ 02K16C056).

References

[1] Lee J, Bagheri B, Kao H-A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems *3*, 2015, p. 18.

[2] Vaidya S, Ambad P, Bhosle S. Industry 4.0 – A Glimpse *20*, 2018, p. 233.

[3] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik. *Reference Architecture Model Industrie 4.0 (RAMI4.0): Statusbericht*, Düsseldorf, 2015.

[4] *Smart Services World – Internet-based Business Services*. Federal Ministry for Economic Affairs and Energy, 2016.

[5] Böhmann T, Leimeister JM, Möslein K. Service Systems Engineering *6*, 2014, p. 73.

[6] Höckmayr B, Roth A. Design of a Method for Service Systems Engineering in the Digital Age, 2017.

[7] Bruhn M, Hadwich K. *Dienstleistungen 4.0: Geschäftsmodelle - Wertschöpfung - Transformation. Band 2. Forum Dienstleistungsmanagement*. Springer Fachmedien Wiesbaden, 2017.

[8] Gitzel R, Schmitz B, Fromm H, Isaksson A *et al*. Industrial Services as a Research Discipline *11*, 2016, 4:1-22.

[9] Polke M, Epple U, Ahrens W, Editors. *Prozeßleittechnik: Mit 8 Tabellen,* 2nd edn. Oldenbourg, 1994.

[10] Schlick J, Stephan P, Greiner T. Kontext, Dienste und Cloud Computing: Eigenschaften und Anwendungen cyber-physischer Systeme, in *atp edition*.

[11] Malleret V. Value Creation through Service Offers *24*, 2006, p. 106.

[12] Furrer O. A customer relationship typology of product services strategies, in *The handbook of innovation and services a multi-disciplinary perspective*, Elgar, Cheltenham [u.a.], 2010, p. 701.

[13] Oliva R, Kallenberg R. Managing the transition from products to services *14*, 2003, p. 160.

[14] Papert M, Pflaum A. *Entwicklung eines Ökosystemmodells für das Internet der Dinge*.

[15] Tantik E, Anderl R. Integrated Data Model and Structure for the Asset Administration Shell in Industrie 4.0 *60*, 2017, p. 86.

[16] Klettner C, Tauchnitz T, Epple U, Nothdurft L *et al.* Namur Open Architecture *59*, 2017, p. 17.

[17] The Open Group. O-PAS™ Standard, Version 1.0: Part 1 – Technical Architecture Overview (Informative), 1st edn. The Open Group, 2019(P190-1).

[18] Choudary SP, Parker G, van Alstyne M. *Die Plattform-Revolution: Von Airbnb, Uber, PayPal und Co. lernen: Wie neue Plattform-Geschäftsmodelle die Wirtschaft verändern*. MITP Verlags GmbH & Co. KG, 2017, Place of publication not identified, s.l.

[19] W3C. Web Services Architecture: W3C Working Group Note 11 February 2004. https://www.w3.org/TR/ws-arch/. Accessed 29 December 2018.

[20] Papazoglou MP. Service-oriented computing: concepts, characteristics and directions, in *Proceedings of the 7th International Conference on Properties and Applications of Dielectric Materials (Cat. No.03CH37417)*, IEEE Comput. Soc, 2003, p. 3.

[21] Josuttis NM. *SOA in der Praxis: System-Design für verteilte Geschäftsprozesse,* 1st edn. dpunkt, Heidelberg, 2018.

[22] Heutschi R. *Serviceorientierte Architektur: Architekturprinzipien und Umsetzung in die Praxis*. Springer, Berlin, Heidelberg, 2007.

[23] Fowler M, Lewis J. Microservices: A definition of this new architectural term. https://martinfowler.com/articles/microservices.html. Accessed 29 December 2018.

[24] Baines TS, Lightfoot HW, Benedettini O, Kay JM. The servitization of manufacturing *20*, 2009, p. 547.

[25] Koenen T, Heckler S, Stündel S. *Deutsche digitale Industrieplattformen – Eine Bestandsaufnahme*, 2018.

[26] Blackwell E, Dhawan R, Gambell T, George K, Schmitz C. *The great re-make: Manufacturing for modern times*, 2017.

[27] Oks SJ, Fritzsche A, Möslein KM. An Application Map for Industrial Cyber-Physical Systems, in *Industrial Internet of Things*, Springer International Publishing, Cham, 2017, p. 21.

[28] Kruchten PB. The 4+1 View Model of architecture *12*, p. 42.

[29] Oks, S.J., Fritzsche, A., Möslein, K.M., 2018. Engineering industrial cyber-physical systems: An application map based method *72*, 1995, p. 456.