



EPiC Series in Computing

Volume 63, 2019, Pages 241–250

Proceedings of 32nd International Conference on
Computer Applications in Industry and Engineering



Towards a Microservice Architecture for the Manufacturing Operations Layer

Christoph Wunck^{1,2,3}

¹ Emden/Leer University of Applied Sciences, Germany

² Department of Computer Science, Iowa State University, Ames, USA

³ OFFIS Institute for Information Technology, Oldenburg, Germany

christoph.wunck@hs-emden-leer.de

Abstract

This paper elaborates on the advantages of migrating legacy IT systems for manufacturing operations to a microservice architecture, which is an important step towards a platform-based ecosystem. Architecture models for manufacturing operations from the literature are evaluated. The different models' strengths are combined towards a common architecture for the factory of the future. Microservices are introduced as a new architectural style for manufacturing operations software.

1 Introduction

Manufacturers all over the world experience constant pressure in four areas: market-related pressure (e.g. customized products, market saturation), economy-related pressure (e.g. globalization of manufacturing, migration towards low cost economies), technology-related pressure (e.g. 3D printing, new materials, big data, security) and environment-related pressure (e.g. energy efficiency, sustainability, product life cycle) [1]. Many manufacturers are small or medium enterprises. For these smaller firms it is difficult to keep pace with the recent developments in *Industrie 4.0* and Smart Manufacturing, which are geared towards flexible manufacturing IT systems that can adapt easily to changing environments. They find it difficult to migrate to new technologies like cyber-physical production systems, machine-to-machine communication, or manufacturing execution and intelligence systems (MES) while sustaining their day-to-day manufacturing business.

Cyber-physical systems (CPS) have their origins in the area of embedded systems. A CPS may be defined as a computer system tightly bound to a mechanical systems or physical process. An important application area of cyber-physical systems is smart manufacturing, where cyber-physical production systems (CPPS) cooperate across all levels of production. CPPS acquire information from their environment, act autonomously, connect to other systems or human operators and respond to internal

and external changes. CPS and CPPS are expected to facilitate the creation of new business models and new services [2].

The functionality of manufacturing operations within a vertical stack of system levels has been standardized by the international standard IEC 62264, also known as ISA-95. Manufacturing execution

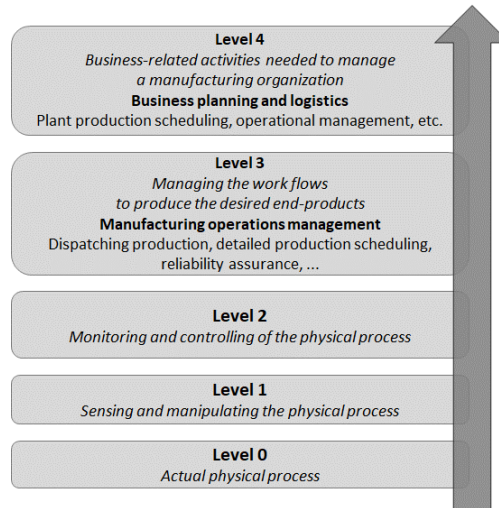


Figure 1: Vertical integration in the multi-level functional hierarchy of IEC 62264 [8]

systems (MES) and other systems targeted to support manufacturing operations reside above the shop floor (level 0, 1, 2) and beneath the enterprise resource planning level (ERP at level 4). The standard defines several categories of information models for the manufacturing operations management layer at level 3.

According to [3], MES will play a central role in manufacturing enterprises' path towards *Industrie 4.0*. On the other hand, MES has not been a widely explored concept in academic research. A migration path towards *Industrie 4.0* for small and medium manufacturers must be both attractive regarding business opportunities and feasible regarding technological challenges. Investing in new technology is a business decision in the first place, and many of today's smart manufacturing technologies lack demonstrating a business case. The recent emergence of platforms and business ecosystems [4], [5] will have a major impact on manufacturing enterprises, as the MES seems to be a suitable candidate for the transition from a monolithic software system into a service- and platform-based ecosystem [6].

This paper elaborates on the advantages of migrating legacy IT systems for manufacturing operations to a microservice architecture, which is an important step towards a platform-based ecosystem. Section 2 evaluates architecture models for manufacturing operations from the literature. Section 3 discusses how the different models may be combined towards a common architecture for the factory of the future. Section 4 introduces microservices as a new architectural style for manufacturing operations software. Section 5 wraps up the conclusions.

2 Reference Architectures and Implementations

Manufacturing operations encompass a large set of activities like detailed scheduling, recipe management, resource management, production execution, work in progress management, production history or quality management, among others. Many manufacturers support their operations using a multitude of customized special-purpose applications and spreadsheet files [7].

Manufacturing Execution Systems (MES) are a type of application software designed specifically to support manufacturing operations. Typically these software systems are huge monoliths developed and marketed by a single vendor. It is difficult or costly for a manufacturer to have custom functions, algorithms or interfaces implemented if these are not contained in the standard set of MES functions.

Industrie 4.0 scenarios assume both smaller lot sizes to manufacture and less stable market conditions that manufacturers need to adapt to. A couple of software architecture patterns and implementations have been proposed by various authors or standardization bodies to accommodate the requirements of flexibility in manufacturing operations. This section discusses the most influential architectures known from the literature.

2.1 IEC 62264 / ISA-95

IEC 62264 is an international standard for enterprise-control system integration, based upon ANSI/ISA-95, which is a standard by the International Society of Automation (ISA). IEC 62264 reflects the hierarchical organization of enterprises by categorizing operations and activities in five levels. This level structure has its roots in the Purdue Enterprise Reference Architecture, developed in the early 1990s by Theodore J. Williams. Level 3 defines manufacturing operations in four areas, namely *production*, *maintenance*, *quality test*, and *inventory*. For each area of operations, four types of models, based on best practices, are defined [8]:

- *Activity models* describe what has to be done at the manufacturing level.
- *Information models* determine what types of entities are active at the manufacturing level, e.g. machines, workers, schedules. Models are elaborated as UML class diagrams.
- *Data flow models* show what types of data are exchanged during activities at the manufacturing level.
- *Data structure models* reveal how the internal structure of data looks like.

The definitions from IEC 62264 can be used to standardize the interfaces between ERP and MES systems, to improve communication in MES projects, and to develop interoperable software [7]. An XML implementation of the IEC 62264 models is given by the Business To Manufacturing Markup Language (B2MML). B2MML consists of a set of XML schemas that implement the data models in the standard. It is maintained by the Manufacturing Enterprise Solutions Association (MESA) XML Committee [9].

2.2 Holonic Manufacturing Systems

The concept of *Holon* was developed in the context of social organizations and living organisms to describe a whole-part-relationship between real-life objects. Holons are both self-contained entities to their subordinated parts, and dependent parts when seen from the inverse direction. H. Van Brussel, P. Valckenaer and others [10] adapted the concept of holons to manufacturing to attain benefits like stability during disturbances, adaptability and flexibility in changing environments, and efficient use of available resources. According to the authors, a Holonic Manufacturing System (HMS) preserves the stability of a hierarchical organization while providing the flexibility of a heterarchy. A holon in a manufacturing system is an autonomous and co-operative building block for transforming, storing, or validating information and physical objects. Therefore a holon consists of an information processing part and often a physical processing part. A holon can be part of another holon. Holons can be modelled by UML objects and class diagrams. They can be part of a generalization hierarchy.

The authors of [10] proposed the HMS reference architecture PROSA (Product-Resource-Order-Staff Architecture). The acronym refers to the four types of holons:

- *Product holons* encapsulate knowledge about products and processes, e.g. bill of materials, process plans, or quality assurance procedures.
- *Resource holons* are abstractions of industrial assets, e.g. factory, machines, tools.
- *Order holons* represent tasks in the manufacturing system. They manage the physical product, its state and logistical information, e.g. customer orders, make-to-stock orders, orders to maintain and repair.
- *Staff holons* assist the aforementioned basic holons in performing their work. While basic holons are responsible for delivering their results and making their own decisions, the staff holons hand out advice. The concept of staff holons facilitates centralized functionality within the system, e.g. production schedules.

Figure 2 shows the three basic holons and their interaction. They exchange *process knowledge* on how to perform operations on resources, *production knowledge* on how to produce products, and *process execution knowledge* on how to execute process instances for certain customers.

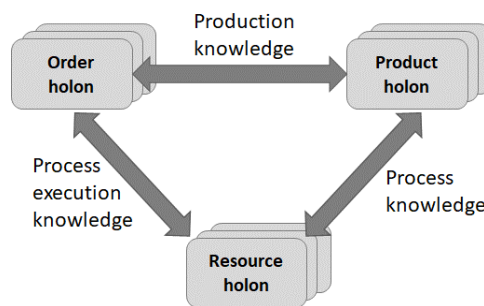


Figure 2: Basic building blocks of a Holonic Manufacturing System [10]

Ongoing research on Holonic Manufacturing Systems targets areas of application and technical implementations. A Manufacturing Execution System based on PROSA was introduced in [11]. The research prototypes have been implemented as multi-agent systems.

The authors of [12] present an adaptive holonic control architecture (ADACOR) with four manufacturing holon classes: *product* holons, *task* holons, *operational* holons, and *supervisor* holons. The prototype was implemented as a multi-agent system based on the Java Agent Development Framework (JADE). An implementation of eye-tracking technology in Holonic Manufacturing Systems is described in [13].

2.3 Multi-Agent Systems

Multi-agent systems have been proposed as the preferred architecture for integrated manufacturing since the late 1990s [14]. A comprehensive summary of agent-based systems for manufacturing is given in [15]. An agent is defined as a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behavior. An agent may have an environment that includes other agents. The community of interacting agents, as a whole, operates as a multi-agent system (MAS). Agent technologies target problems like agent behavior and decisions, interaction, organization, division of labor, coordination, supervision. Holonic Manufacturing Systems are viewed as a special kind of MAS.

Many academic implementers use the Java Agent Development Framework (JADE) as their middleware platform. The software architecture is based on cooperating Java Virtual Machines. The communication between agents relies on Java Remote Method Invocation (RMI) and IIOP [16]. Agent

services can be exposed as web services (WSDL). In-depth technological and architectural details about this framework are presented in [17].

A multi-agent implementation based on JADE for the holonic control of a manufacturing cell is presented in [18]. The authors conclude that MAS is a useful approach for the implementation of holonic architectures, as software agents and holons share several similarities. The authors of [19] focus on the comparison of multi-agent systems and IEC 61499 function blocks.

2.4 Service-oriented Architectures

Service-oriented architectures for industrial applications have been proposed by [20] in the context of the European Union FP7 project IMC-AESOP. The authors envision the factory of the future as a composition of services of varying complexity, and composed by other (potentially cross-layer) services, as depicted in Figure 3. Services may be hosted either on field devices, local/edge servers or in the cloud.

The flat information-based architecture of the future coexists with the traditional hierarchical view from IEC 62264. Next generation industrial applications can rapidly be composed by selecting and combining new information and capabilities offered as services in the cloud. The collection of cloud services is targeted towards supporting IEC 62264 operations and activities.

The authors of [20] have elaborated several important requirements that need to be observed when implementing a service-oriented IT system for manufacturing operations support. In reference to architectural considerations, the following non-functional requirements are of particular interest:

- *Backward and forward compatibility*: Evolving infrastructure must not break existing functionality.
- *Combinability of services and tools*: Software applications will be created by a reuse-based software process according to [21].
- *Dynamic service discovery*: New devices or services announce their presence, while other components get aware of new capabilities.
- *Real-time interaction*: Technologies must meet the performance requirements of real-time interactions. Soft real-time essentially means low latency in the processing pipelines.
- *Technology-agnostic infrastructure evolution*: Future systems need to be updated to evolving technologies and simultaneously maintain their functionality.

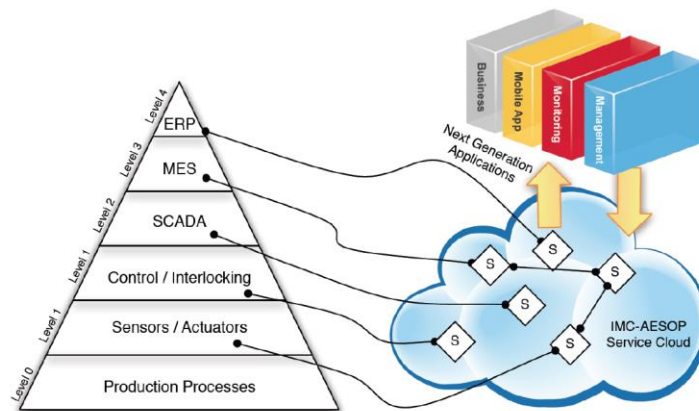


Figure 3: Cloud-base composition of cyber-physical services [20]

- *Interoperability and open exchange formats:* Systems and services will have to communicate vertically and horizontally between systems at remote sites.
- *Mobility support:* Future systems need to be accessed via mobile devices, must support mobile assets in manufacturing and facilitate the migration of services.
- *Infrastructure Services:* Service developers should rely on a comprehensive middleware stack to not have to code software from scratch.
- *Scalability:* It should be easy to scale the system to new usage profiles, balance loads or deliver a defined quality of service.

Details regarding the implementation of the service-oriented architecture can be drawn from the IMC-AESOP project website [22], e.g. services are implemented as web services using technologies like HTTP, XML, SOAP, and WSDL.

2.5 Reference Architecture Model Industrie 4.0 (RAMI 4.0)

RAMI 4.0 was developed in Germany by the *Plattform Industrie 4.0* consortium and standardized internationally in 2017 [23]. RAMI 4.0 references a couple of preexisting standards and enforces their application. The standard focusses on the cooperation and collaboration of technical assets. An asset is defined as any tangible or intangible item possessing a value for an organization. Assets can be as small as a screw and as extensive as an MES or ERP system. Assets have a life cycle and must be described in three different views or dimensions.

An *architectural view* describes the structure of an asset in terms of layers, similar to stacked layers of software systems. Interactions occur between adjacent layers. The top layer describes the asset's business context, the bottom layer corresponds to the real physical asset.

A *life cycle view* describes an asset from design, creation, usage and value generation, until phase-out and disposal.

A *hierarchy view* describes an asset's function in the context of a manufacturing site. The hierarchy is modeled after IEC 62264 and extended downwards to include the product itself as a value-adding part in the manufacturing value chain and upwards to include the connected world with cooperations between factories.

The standard illustrates the three views in the form of a three-dimensional layered cube, as depicted in Figure 4. The author of this paper has indicated the scope of manufacturing operations corresponding to IEC 62263 level 3 by red dotted lines.

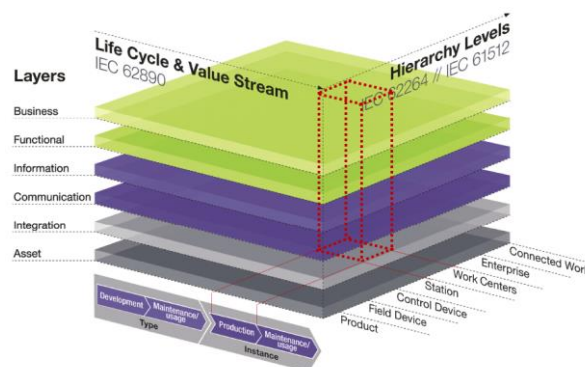


Figure 4: Scope of manufacturing (level 3) operations within RAMI [23]

The RAMI architecture model envisions manufacturing sites as a collection of so-called *Industrie 4.0 Components*. These are worldwide uniquely identifiable entities made of an asset and a software representation called *administration shell*. Industrie 4.0 components can be composed from other components and can themselves be used to form more complex components. A component's attributes and state may be queried via software interfaces in the administration shell.

RAMI is designed to resolve basic interoperability issues in manufacturing and thus lays the foundation for the development of complex cyber-physical production systems (CPPS) that autonomously exchange information, trigger actions, and control each other independently [2]. The authors of [24] elaborate on the concept of an adaptive MES querying the administration shell of relevant assets in order to collect enough information about the manufacturing environment to adapt itself to changes.

3 Discussion

The previous section presented short summaries of different architecture models for manufacturing operations from the literature. This chapter elaborates on how the respective strengths of these models can be combined to complement each other.

IEC 62264 partitions the application domain of manufacturing operations into smaller blocks, thus supporting the definition and validation of software requirements by domain specialists. The hierarchical model of IEC 62264 maps the division of labor prevalent in many successful enterprises. The level system has demonstrated its usefulness for the last more than 20 years and will be a part of coming solutions for the factory of the future, due to being one of the major dimensions of the RAMI 4.0 model. RAMI 4.0 is the foundation standard to ensure interoperability between assets. It will have a profound influence on the software landscape in industrial automation. However, many aspects important to smart manufacturing applications are not addressed by this very basic standard and still left to the decision of implementers.

The service-oriented architecture (SOA) proposed by [20] is an important step towards more architectural flexibility. Though the authors primarily focus on cloud-based services, the concept of SOA can easily be applied to local services on premises. The services offered might correspond to parts of an administration shell to access Industrie 4.0 components. The flexibility to design applications and to run them on a variety of software platforms will probably be a major incentive for manufacturers to abandon their customized legacy software and migrate to service-oriented architectures. The technology to implement SOAs varies in the course of time. Web services based on XML, SOAP and WSDL recently have lost some acceptance in the software engineering community due to low performance and high complexity in favor of microservices [25].

Service-oriented architectures are a technical requirement to create economic platform ecosystems. These will create completely new business opportunities for startups, small and medium enterprises both from manufacturing and ICT. A platform *owner* provides the infrastructure and rules for a marketplace that brings together *producers* and *consumers* of services [5]. *Providers* grant access to a platform via interfaces. In the manufacturing industry, machines, operators, or management might be both producers and consumers of services, depending on their respective role within the processes. Providers might be devices like retrofitting devices, mobile phones, or networking interfaces like OPC UA. Platform owners might be today's vendors of MES, newcomers from related industries, or open source solutions without explicit ownership.

Holonic manufacturing systems (HMS) and multi-agent systems (MAS) have much in common. As mentioned before, many existing HMS implementations use an MAS approach. MAS oftentimes are implemented using the JADE middleware. Holonic manufacturing strives for highly autonomous modes

of operation, which might collide with regulations in manufacturing regarding quality certification, safety procedures or legislation. HMS/MAS however might contribute to factories of the future in certain unregulated areas, where better alternatives to human decisions are welcome. In the context of a service-oriented architecture, HMS/MAS might even provide sophisticated services like planning and analytics, without interfering with highly regulated processes. Many technical implementations of MAS are bound to the JADE middleware, which is described by its maintainers as a niche technology far from mainstream software engineering [17]. This might be one of the obstacles that keeps small and medium manufacturing enterprises away. As the JADE runtime can be embedded into a hosting Java application, MAS applications based on JADE could be easily integrated into an SOA landscape.

4 Microservice Architecture

Microservices are the second generation of service-oriented architectures. They emphasize the development of highly maintainable and scalable software by decomposing large systems into sets of independent services with distinct business capabilities [25]. While the first generation of web services focused on complex technologies like syntactic service descriptions, discoverability and message contracts using XML Schema and Web Service Description Language (WSDL), the second generation aims to remove complexity in favor of simple functionalities. [25] gives an in-depth overview of the evolution of distributed and service-oriented computing from an academic viewpoint.

The authors of [25] define a monolith as a software application whose modules cannot be executed independently, while a microservice is considered a cohesive, independent process interacting via messages. They confront six major issues of monoliths with corresponding microservice solutions:

1. *Evolution*: Large monoliths are difficult to adapt to new requirements. Microservices implement a limited functionality with a small codebase, keeping maintenance costs low.
2. *Dependencies*: Monoliths depend on numerous external libraries spread over different places by the operating systems. These dependencies may lead to versioning conflicts. Microservices can gradually evolve to a new version, while a previous version of the same service remains available as long as required.
3. *Deployment*: Deploying small changes in a monolith requires restarting the whole application. Microservices require only themselves to be restarted, resulting in short redeployment times.
4. *Runtime environment*: Deployment environments for monoliths cannot allow for all requirements of all modules equally. Microservices are run from containers providing an environment optimally tailored to the microservice's task.
5. *Scaling out*: Increasing load on a monolith is compensated by distributing the load among multiple instances of the whole monolith, which might not address the load problem efficiently. Multiple instances of microservices can address an increasing load in finer detail.
6. *Technology lock-in*: Evolution of a monolith is bound to the original language and technology. Microservices can migrate to new technologies any time during evolution, as long as the communication interfaces remain stable.

The services proposed by [22] as well as the service export mechanism of the JADE middleware belong to the first generation of web services requiring monolithic application servers for their execution. Due to their advantages over first-generation web services the author of this paper envisions that microservices will be the technological foundation of service-oriented architectures for manufacturing operations. The requirements posed in section 2.4 on SOA in manufacturing settings can be met by microservice implementations:

- *Backward and forward compatibility*: Microservices are designed to smoothly support both evolution of implementations and interfaces.
- *Combinability of services and tools*: This is a general property of SOA and therefore applies to microservice architectures alike.
- *Dynamic service discovery*: Service discovery as such is not part of microservice technology, but is supplied by middleware stacks like *Vert.x*, *Spring Boot*, and others.
- *Real-time interaction*: Microservices are designed for low latency.
- *Technology-agnostic infrastructure evolution*: Microservices avoid technology lock-in and allow easy migration to technological innovations.
- *Interoperability and open exchange formats*: Message exchange is based on open protocols and formats like Hypertext Transfer Protocol (HTTP), XML and JSON.
- *Mobility support*: Current patterns for mobile application development expect microservices at the backend. Microservices are mainstream technology for this kind of software.
- *Infrastructure Services*: A choice of middleware stacks is available.
- *Scalability*: Microservices have been designed explicitly with respect to scalability.

5 Conclusions and Future Work

This paper examined several known architectures for information systems in manufacturing: Hierarchical Enterprise and Control Systems, Holonic Manufacturing Systems, Multi-Agent Systems, Service-Oriented Architecture (SOA), and Reference Architecture Model Industrie 4.0. The respective strengths of each architecture was acknowledged. It was shown that an SOA will likely be the foundation for the factory of the future, and aspects of each architecture under consideration could be accommodated within an SOA. Finally the microservice architecture style was proposed to remedy various deficiencies of earlier SOA middleware implementations. Future work will be directed to evaluate a use case based on the injection molding monitoring application presented in [14].

References

- [1] M. Abramovici, F. Bellalouna and J. C. Boebel, „Towards adaptable industrial product-service systems,“ in *Proceedings of the 2nd CIRP IPS2 Conference 2010*, Linköping, Sweden, 2010.
- [2] L. Monostori, B. Kádár, T. Bauernhansl, et al., „Cyber-physical systems in manufacturing,“ *CIRP Annals - Manufacturing Technology*, pp. 621-641, 65 2016.
- [3] S. Mantravadi and C. Møller, „An Overview of Next-generation Manufacturing Execution Systems: How important is MES for Industry 4.0?,“ *Procedia Manufacturing*, 2019.
- [4] R. Kapoor, „Ecosystems: broadening the locus of value creation,“ *Journal of Organization Design*, 2018.
- [5] M. W. Van Alstyne, G. G. Parker and S. P. Choudary, „Pipelines, Platforms, and the New Rules of Strategy,“ *Harvard Business Review*, April 2016.

- [6] C. Wunck and S. Baumann, „Manufacturing Execution Systems (MES) – the next platform ecosystem,“ *EDSI Annual Meeting*, Udine, Italy, June 4, 2018.
- [7] B. Scholten, *MES Guide for Executives*, Research Triangle Park, NC, USA: International Society of Automation, 2009.
- [8] IEC 62264, „Enterprise-control system integration,“ 2015.
- [9] MESA, „Business To Manufacturing Markup Language (B2MML),“ [Online]. Available: <http://www.mesa.org/en/B2MML.asp>. [Accessed: July 07, 2019].
- [10] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts and P. Peeters, „Reference architecture for holonic manufacturing systems: PROSA,“ *Computers in Industry*, pp. 255-274, 37/1998.
- [11] P. Valckenaers and H. Van Brussel, „Holonic manufacturing execution systems,“ *CIRP annals*, pp. 427-432, 1/2005.
- [12] P. Leitão and F. Restivo, „ADACOR: A holonic architecture for agile and adaptive manufacturing control,“ *Computers in industry*, pp. 121-130, 2/2006.
- [13] J. Niemann, A. Basson, et al., „Implementation of Eye-Tracking technology in Holonic Manufacturing Systems,“ *Procedia - Social and Behavioral Sciences*, pp. 37-45, 238/2018.
- [14] C. Wunck, „Implementation of Mobile Event Monitoring Agents for Manufacturing Execution and Intelligence Systems Using a Domain Specific Language,“ in *Proceedings of the 2016 IEOM Conference*, Kuala Lumpur, Malaysia, 2016.
- [15] L. Monostori, J. Váncza and S. Kumara, „Agent-Based Systems for Manufacturing,“ *CIRP Annals - Manufacturing Technology*, 1/2006.
- [16] F. Bellifemine et al., „General Questions about Jade,“ [Online]. Available: <https://jade.tilab.com/support/faq/>. [Accessed: July 07, 2019].
- [17] F. Bellifemine, G. Caire, et al., „JADE: A software framework for developing multi-agent applications,“ *Information and Software Technology*, pp. 10-21, 50 2008.
- [18] K. Kruger and A. Basson, „JADE Multi-Agent System Holonic Control Implementation,“ 2/2018. Available: http://academic.sun.ac.za/mad/Papers/KrugerBasson_HolonicControlImplementationUsingJadeMAS_20180208.pdf. [Accessed: July 07, 2019].
- [19] K. Kruger and A. Basson, „Multi-Agent Systems vs IEC 61499 for Holonic Resource Control in Reconfigurable Systems,“ *Procedia CIRP*, pp. 503-508, 7/2013.
- [20] S. Karnouskos, A. W. Colombo, et al., „A SOA-based architecture for empowering future collaborative cloud-based industrial automation,“ in *38th Annual Conference on IEEE Industrial Electronics Society (IECON)*, Montreal, Canada, 2012.
- [21] I. Sommerville, *Software Engineering*, Pearson Education, 2016.
- [22] IMC_AESOG Project, [Online]. Available: <http://www.imc-aesop.org/index.html>. [Accessed: July 07, 2019].
- [23] IEC PAS 63088, „Smart manufacturing - Reference architecture model industry 4.0 (RAMI 4.0),“ 2017.
- [24] J. Kallisch and F. Oppenheimer, "Adaptive manufacturing execution systems (AMES): Best practices for implementations in small and medium-sized businesses," in *Proceedings of the 10th Annual European Decision Sciences Conference (EDSI)*, Nottingham, UK, 2019.
- [25] N. Dragoni, S. Giallorenzi et al., "Microservices: yesterday, today, and tomorrow." in *Present and ulterior software engineering*, pp 195-216, 2017.