

# Software architectures in smart manufacturing: Review and experiences

Zeljko Stojanov<sup>1</sup>, Dalibor Dobrilovic<sup>1</sup>, Gordana Jotanovic<sup>2</sup>, Dragan Perakovic<sup>3</sup>, Goran Jausevac<sup>2</sup>, Vladimir Brtka<sup>1</sup>

<sup>1</sup> University of Novi Sad, Technical faculty "Mihajlo Pupin", Serbia

<sup>2</sup> University of East Sarajevo, Faculty of transport and traffic engineering, Republic of Srpska, Bosnia and Herzegovina

<sup>3</sup> University of Zagreb, Faculty of Transport and Traffic Sciences, Croatia

E-mail: [zeljko.stojanov@uns.ac.rs](mailto:zeljko.stojanov@uns.ac.rs)

**Abstract.** Cyber-physical systems based on heterogeneous and distributed devices, applications and services are the core of smart factories. Smart manufacturing systems are highly dependent on software applications and services that enable integration of data from different and heterogeneous sources, as well as support for control and management processes. Development and implementation of specific architecture styles and patterns in smart industrial settings is essential for their performance. Several software architecture styles used in Industry 4.0 environments are discussed and illustrated with examples from literature. Our experience with a prototype of a smart sensor-based layered architecture is presented and discussed. Further work will be directed towards development of service oriented architectures and reengineering method for old fashioned industrial settings.

## 1. Introduction

The term Industry 4.0 (I4.0) was introduced by industry leaders from Germany in last ten years, and it denotes the Forth Industrial Revolution or introduction of "smart manufacturing" in industrial settings. Several different terms have been used interchangeably for Industry 4.0, like Industrial Internet, Connected Enterprise, SMART Manufacturing, Smart Factory, Manufacturing 4.0, Internet of Everything, or Internet of Things for Manufacturing. Industry 4.0 production systems are based on cyber-physical systems with the extensive use of Internet of Things and Services. This enables creating networks of cyber-physical components that incorporate entire manufacturing process and create a smart production environment [1]. Fundamental principles of Industry 4.0 are [2]:

- *Use of Internet.* Internet provides inexpensive channels for connecting machines, devices, sensors and people. In addition, it can be used for creating new functions and features in industrial settings and collecting large amounts of data.
- *Production flexibility.* It is reflected through real-time functioning and minimization of setup-time, support for demand of personalization and mass customization, and fast and cheap prototyping.

- *Communication and virtualization of cyber-physical systems.* Development of a common communication framework is essential for connecting machines, sensors and products in industrial settings, which enables defining their functionalities and treating them as virtual computational entities.

Information and communication technologies form the infrastructural foundation for industrial cyber-physical systems. The most important technologies for Industry 4.0 cyber-physical systems are [3]:

- *Internet of Things (IoT) and related technologies.* IoT is based on network infrastructure composed of connected devices, communication and networking, and technologies for information processing. The most important technologies that enable IoT are Radio-Frequency Identification (RFID) and Wireless Sensor Networks (WSN).
- *Cloud computing.* Cloud computing provides effective solutions for services and resources distribution and sharing, leading to optimization of data exchange and increased effectiveness of manufacturing resources. It enables the achievement of high performance with modularization and service-orientation in cyber-physical systems.
- *Industrial and enterprise application integration.* Efficient integration and coordination are critical for cyber-physical systems composed of heterogeneous components. It includes integration of heterogeneous data sources, processes, applications, platforms and standards. The common trend in integrating heterogeneous systems assumes the use of service-oriented architecture (SOA).

Dynamic development of smart manufacturing is accompanied by the development of standards and reference architecture models that aim to provide the best practice recommendations. Some of the most important endeavors are *The Industrial Internet of Things Reference Architecture (IIoT)* [4] proposed by Industrial Internet Consortium, *ISO/IEC 30141:2018 Internet of Things (IoT) - Reference Architecture* proposed by ISO/IEC technical committee, and *IEC PAS 63088:2017 Smart manufacturing - Reference architecture model industry 4.0 (RAMI4.0)* proposed by International Electrotechnical Commission (IEC). These standards and reference models have led to development and implementation of specific architecture styles and patterns in industrial settings. Fraile et al. [5] presented Industrial Internet Integrated Reference Model (I3RM), which is based on the integration of features of prominent reference models for IIoT systems. The proposed model uses business, usage, functional, and implementation viewpoints as a framework for defining the system architecture.

Li et al. [6] systematized the standards for I4.0 proposed by ISO, IEC, ITU, IEEE, and other international standard development organizations. This systematization includes analysis and comparison of smart manufacturing architectures, proposal of a reference model for smart manufacturing standards development, and finally development of a framework for standards that enable classification of standards. According to the proposed framework, standards related to smart manufacturing can be divided in the following groups: smart design standards, smart production standards, business operation and management standards, system integration standards, and fundamental technologies and supporting environment standards (this group includes software engineering standards).

Software applications and services in Industry 4.0 are used in manufacturing environment and for managing business processes. In manufacturing environment, software applications and services are used for data acquisitions through sensors, control and optimization of manufacturing processes, and management of operational and process data. Software applications and services are also used for planning and management of business processes, inter-company logistics and integration of business processes and manufacturing processes. Integration of software services and applications in smart Industry 4.0 networked environment enables

allocation and deployment of all manufacturing and business processes in App Stores style model [1]. Since integration of data from different sources and different software technologies is the core of smart industrial systems, special attention should be paid to design of software architecture and selection of appropriate software and IT technologies [7]. Software systems that manage and control complex and heterogeneous smart manufacturing systems should adopt novel and architectural styles and patterns [8]. Several software architecture patterns have been used in complex IT and industrial systems, such as layered architecture, event-driven architecture, microkernel architecture, service-oriented architecture, microservices architecture, or space-based architecture [9, 10].

This paper is structured as follows. The second section outlines a literature review of software architectures in smart manufacturing settings, accompanied with the summary of identified architectural patterns and challenges need to be solved. The third section presents our experience with layered sensor-based software architecture in smart manufacturing. Conclusions and future work directions are presented in the last section.

## **2. Software architectures in smart manufacturing**

This section presents a literature review of studies focused on using software architecture patterns (styles) in smart manufacturing systems, followed with summary on used software architecture patterns and identified challenges in the analyzed studies.

The literature review is based on the studies selected through search on Google Scholar, a freely accessible web search engine for scholarly literature. Two searches were performed by using the following combinations of keywords: (1) "software architecture" and "Industry 4.0", and (2) "software architecture" and "smart manufacturing". The range for years of the published studies is limited from 2010 to 2020. Since Google Scholar returned about 3,500 selected literature sources for two searches, the review was constrained on the first 20 sources in both searches, and examination of their references. The inclusion of the selected studies is based on the following criteria: (1) the full paper is available, (2) the study is published in refereed journal or in proceedings of international conference, (3) the study is written in English, and (4) the study explicitly deals with software architecture in smart manufacturing settings.

### *2.1. Literature review*

Based on a literature review and semi-structured interviews with experts Vater et al. [11] identified the requirements for IT architecture for prescriptive automation in smart manufacturing systems. On the basis of the identified requirements, solution components are defined. The main categories of identified requirements are: control, data acquisition, data processing, connectivity and data storage. A reference architecture is derived from identified requirements and solutions. The reference architecture is based on edge computing solutions for operational control (field and control layers) and cloud computing processing resources for recalculating model. The proposed reference model has the following layers: sensor network, control network and compound network with edge control devices, and information network with information gateway and model recalculation in the cloud.

Kavakli et al. [7] presented a layered component-based architecture of DISRUPT smart industrial system for decision making in manufacturing enterprises through identification and handling of events that could disrupt operations in the system. The components of DISRUPT are organized into the following five layers: (1) Physical Layer comprising of factory assets and Manufacturing Information Systems (MIS) that collects data from assets, (2) Virtualisation Layer provides interface to Physical Layer and aggregate data from MIS, (3) Operational Layer is used for data analysis, disruptive event detection, and identification of trends and patterns, (4) Decision Layer models knowledge derived from data and provides optimization and simulation facilities, and (5) Visualisation Layer provides the interface between system and end users. For

system architecture modeling, three architectural viewpoints are used: (1) Logical viewpoint describes high level architecture in terms of components by using UML class and component diagrams, (2) Informational viewpoint presents system dynamics and exchange of information by using UML informational flow and sequence diagrams, and (3) Physical viewpoint presents allocation of system components to different software containers, execution environments and physical devices by using UML deployment diagrams. The main contribution of the paper is the proposed model of a software architecture that supports managing disruptive events in smart manufacturing.

The Line Information System Architecture (LISA) is an event-driven and service-based architecture that supports rapid and flexible integration of smart services and devices into factory infrastructure [12]. It simplifies hardware changes and integration of new smart services, and provides support for continuous improvements of control and visualization of information. LISA is based on loose coupling of services and devices, and on flexible message structure for integration. The core of LISA system is an Enterprise Service Bus (ESB) that supports message routing between distributed applications and services. LISA message format enable transformation of events into usable information within information system. The architecture has been evaluated on real industrial data in automotive industry. Implementation results indicate time savings during the production system upgrade or when introducing new products.

Cloud based integrated enterprise information system that maps integration of many enterprise information systems (EIS) into private cloud is presented in [13]. Proposed three-layer service-oriented architecture enables integration of hybrid wireless networks into EISs. The layers are: (1) the backend layer that is responsible for business applications that access EIS at the level of web services, (2) the middle layer that is responsible for message transformation, service deployment in the network, and service deployment in the cloud, and (3) the front-end layer that includes infrastructure devices, wireless sensor networks, mobile networks and industrial monitoring networks.

Cai et al. [14] proposed configurable information service platform for IoT applications. Configurable and open software platform enables management of the whole product life-cycle through integration of heterogeneous and distributed product information. The system is based on resource-oriented architecture with IoT applications and RESTful services as the main components. The framework includes life-cycle, product structure, and information dimensions. Life-cycle management dimension includes the following life-cycle stages: design, production, assembling, utility, maintenance and recycling. Product dimension relates layers such as product, components, and parts with unique IoT identification. Information dimension relates to information stored in distributed environment.

Papazoglou et al. [15] presented a reference architecture based on service oriented architecture that enables integration of enterprise and control systems in smart manufacturing. A generic industry-shared platform was implemented by using common software modules. Proposed reference architecture was illustrated through scenario in smart automotive manufacturing for managing all stages of vehicle assembly.

The architecture for Evolvable Assembly Systems (EAS) within smart manufacturing is presented by Chaplin et al. [16]. The main objective of the proposed architecture, and modeling approach is to enhance evolvability of manufacturing environment due to changes in product, processes, or market. The approach is based on the four phases of EAS cycle: (1) (Re)Configuration, (2) Operation, (3) Monitor, and (4) Definition and Adaptation-External and Adaptation-Internal. The general architecture of the proposed EAS corresponds to the phases of EAS cycle. The key modules in the proposed software architecture are: Agent Environment, Definition module (allow users to create definitions for resources, products,), Reconfiguration module (tracks the internal structure of the production line), Monitoring module (semantically enhanced part-tracking database), and Translation module (provides interfaces

towards manufacturing resources). The core part of this architecture is the Agent Environment in which are defined intelligent agents as autonomous software components. The applicability of the proposed architecture was shown through a demonstration by using a manufacturing prototyping platform for customisable pharmaceutical products.

Wan et al. [17] proposed the software defined Industrial Internet of Things (IIoT) architecture for flexible management of network resource for I4.0. The architecture comprises of three layers: (1) *Physical Infrastructure Layer* that contains all devices, sensor platforms, fieldbus control network, robot network, core and cloud networks, (2) *Control Layer* that implements interfaces between physical and application layers, and (3) *Application Layer* that contains APIs for designing software applications for equipment fault monitoring, equipment utilization rate monitoring, and product processing status monitoring. The prototype platform was designed for testing proposed software defined IIoT architecture. Result shows improvement of equipment utilization rate and reduction of the energy consumption, compared to traditional industrial schema.

Lucas-Estañ [18] presented a heterogeneous, hierarchical and multitier communication management architecture that supports ubiquitous, flexible and reliable connectivity and efficient data management in Industry 4.0 scenarios with multiple heterogeneous applications and services. The proposed architecture has been developed within H2020 AUTOWARE project. Hierarchical architecture approach is based on a central orchestrator, a component that coordinates decentralized communication and data management functions. Flexibility, scalability and architectural adaptability are enabled through use of RAN (Radio Access Network) Slicing and Cloud RAN technologies.

Microservice oriented architecture (MOA) for I4.0 applications is presented in [19]. Proposed flexible and interoperable architecture improves deployment and composition of services by using Molecular framework. In this framework all services are equal. The services are distributed into two hierarchical levels: Business/Processes microservices and Infrastructure microservices. Service Broker is the main component in the proposed architecture, responsible for configuring microservices. Experimental evaluation of developed MOA was conducted in process control of DC motor velocity. In addition to standardized features of MOA, service monitoring and standard API for external communications and services' communications are provided. Experimental results confirm usability of proposed architecture for process control and automation in the context of I4.0.

Microservice composition based on service orchestration for process control applications in I4.0 context is presented in [20]. Micro service oriented architecture uses Molecular framework for performing microservices' orchestration. Central component in orchestration controls and coordinates the calls of all services in request-response way. Experiment with the control of closed loop of a pipe pressure in an industrial plant is presented. Experiment results proved that proposed architecture is flexible, scalable and ease of implementation.

Cao et al. [21] proposed an ontology-based holonic event-driven architecture (Oh-EDA) for autonomous networked manufacturing systems. The event-driven architecture organizes event services in a holonic manner, which improves the security management and robustness of event-driven systems. The services can be provided by different organizations and can be autonomously configured and integrated in a plug-and-play manner. The knowledge of services is modeled via event-driven ontology, which enables interoperability of services. Oh-EDA was developed as a conceptual model, in which all concepts and their relationships are defined. The main concepts in the model are: Organization, Service, Event, EventService, EventServiceManager, EventScope, EventType, etc. Demonstration of effectiveness, scalability and reliability of the approach based on Oh-EDA was performed by implementing a testbed using Java related technologies.

## *2.2. Analysis of software architecture styles*

Smart manufacturing systems are characterized by high diversity of systems' architectures and heterogeneity of used technologies and components. Software systems are used in all parts of these systems, from sensing elements, communication elements, to high level management of distributed manufacturing. This is reflected to architecture styles of software elements in smart manufacturing systems, which tend to follow the architecture of the whole system.

Preliminary literature review revealed several software architecture style in smart manufacturing. The most important styles are layered architecture, event-driven architecture and service oriented, although some studies reported mixing of different architecture styles.

Layered software architecture is the consequence of dividing manufacturing system in different layers depending on the used equipment and processes. The number of layers depend on the context, but in most cases sensing, middleware and user layer can be distinguished. Layered software architecture was used in [7, 11, 13, 17, 18].

Event driven architecture reflects the nature of manufacturing systems in which several events may occur, and software should provide optimal actions. Event driven architecture are reported in [12, 21].

Service-oriented architecture (SOA) organizes software components via reusable services that use common communication standards. In this group are classified microservices architecture, in which microservices are software services that are loosely coupled, independent and easy maintainable and testable. SOA is reported in [15], while microservices architecture is reported in [19, 20]. SOA combined with event-driven architecture is reported in [12], while SOA combined with layered architecture is reported in [13].

In addition to mostly user architecture styles, in [14] the use of resource oriented architecture is reported, while in [16] the use of evolution life-cycle based architecture is reported.

## *2.3. Identified challenges*

High complexity of manufacturing systems and dynamics of the global market pose several challenges to researchers and practitioners in designing and implementing smart production systems based on extensive use of heterogeneous and distributed software components. Some of the challenges are common for many manufacturing systems, such as integration of services and data from heterogeneous sources, while some challenges are context specific (driven by specific problem or requirements).

Based on the presented literature review the most cited challenges are aggregation and management of heterogeneous data [12, 14, 18], integration of the different technologies by using shared services [18–20], control of low-level processes and applications [11, 12], and technology standardization [17, 21].

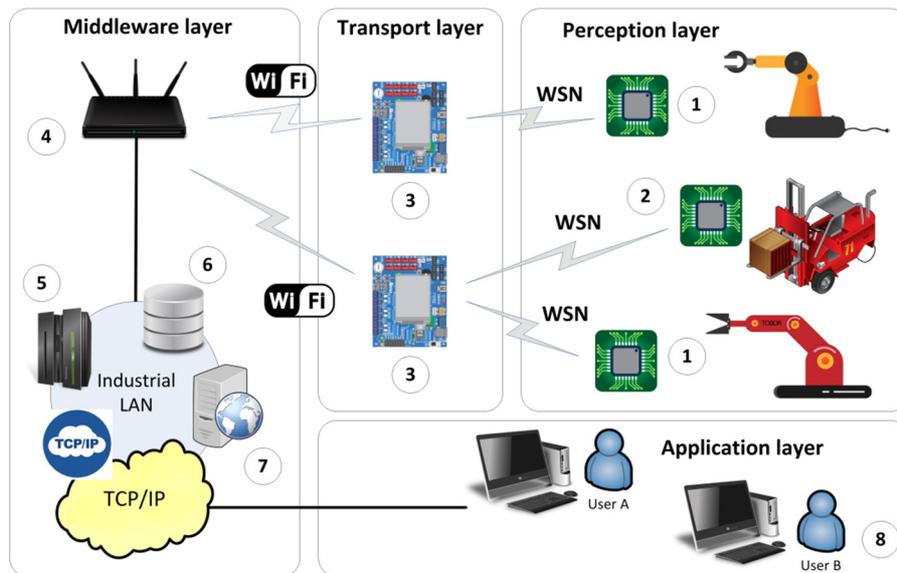
Some of the context specific challenges are support for decision making [7], development of configurable and open software platform [14], enabling dynamic and controlled collaboration across manufacturing systems [15], identification of requirements for system architecture based on experts opinions [11], and optimized reconfiguration of smart manufacturing [16].

## **3. Layered sensor-based architecture for smart manufacturing**

The system for monitoring the industrial environmental conditions in manufacturing settings is presented in Figure 1. The system was developed as a prototype started as student project [22] and it is based on open-source software and hardware components. The presented prototype has sensor-based layered architecture.

The elements of the systems are sensor nodes annotated with (1) and (2). These sensor nodes are based on open-source hardware micro-controller boards such as Arduino UNO Rev3, Arduino MEGA, TI MSP-EXP430G2 LaunchPad Development Board and TI MSP EXP432P401R LaunchPad Development Board. The sensor nodes have temperature, humidity, dust, gas and

smoke sensor in order to monitor the working conditions in the production facilities. The fixed sensors nodes are designed for machines and mobile nodes are designed for moving objects such as forklifts, carts, etc. The nodes have wireless connectivity module with support of various short range wireless technologies such as ZigBee, Bluetooth Low Energy, and Sub-Gigahertz RF technology. The nodes are placed in Perception layer of the system. In the transport layer, the Wi-Fi and short range wireless technology gateways (3) are placed. They convert communication to TCP/IP protocol and forward data to the core of the system. The core of the system, the Middleware Layer, is reachable via Access Points (4), where the data are further transferred to the wired Industrial LAN. In the core of the Industrial LAN the data analytics server (5), database (6) and the web server (7) are located. The data are available to the end users, and control center (8), with application provided with the Application layer. The communication between the Transport layer gateway and server is based on UDP protocol. The architecture of the system is a little bit out of date, and the plan is to modernize the system with the inclusion of the messaging protocols such as MQTT and Mosquitto MQTT broker for the communication between Gateways, Data Analytic server, Data storage and Web Server according to the experiences from the research [23].

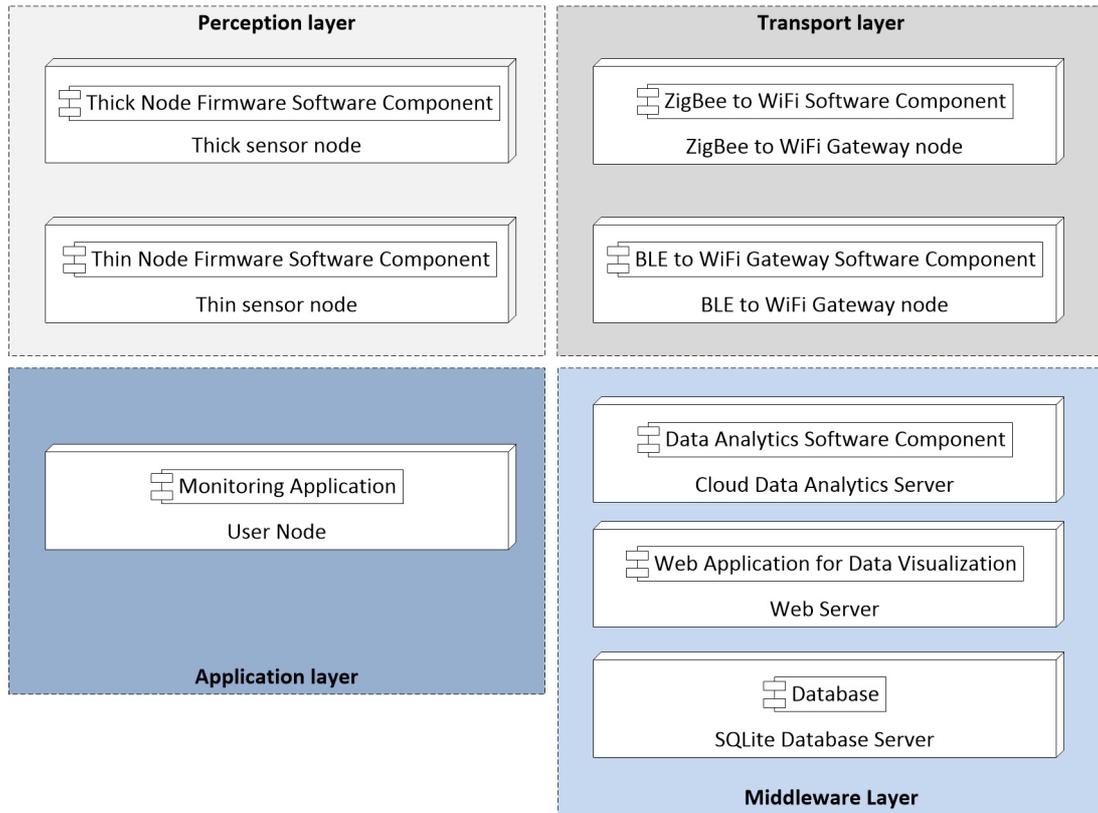


**Figure 1.** Layered architecture of smart manufacturing system

### 3.1. Architecture of software part of the system

Software architecture with physical deployment of software components on nodes is presented in Figure 2. In order to achieve a better overview of the physical arrangement of the software components, the layers determined in Figure 1 are shown.

In *perception layer* are placed thick sensor stations based on TI MSP EXP432P401R, and thin sensor stations based on TI MSP-EXP430G2. Thick sensor stations collect and process data from dust, sound, light, temperature (LM35) and gas (MQ-135) sensors. Thin sensor stations collect and process data temperature/humidity (DHT22) and gas sensor (MQ-135) sensors. Each sensor station has a firmware software component (*Thick Node Firmware Software Component* and *Thin Node Firmware Software Component*) with start-up and loop sequences written in C programming language.



**Figure 2.** Software physical architecture

*Transport layer* contains wall mounted gateways between WSN and LAN based on CC3200-LAUNCHXL. Each gateway contains firmware software component written in C programming language, which is responsible for establishing communication and transmitting data to the middleware layer of the system. The first gateway contains a software component that enables transmission of the data from ZigBee to WiFi (*ZigBee to WiFi Software Component*). The second gateway contains a software component that enables transmission of the data from Bluetooth Low Energy (BLE) to WiFi (*BLE to WiFi Gateway Software Component*).

*Middleware layer* is the core of the system, with the servers placed in the cloud. The first server is *Cloud Data Analytics Server*, which hosts *Data Analytics Software Component* responsible for receiving data from transport layer and processing them. This component is implemented as Blynk Server, an Open-Source Netty based Java server. It is responsible for calculating trends, predictions and preparing reports. Web server is implemented as The Apache Tomcat, Java open source web server, which is the container for the software component for data visualization (*Web Application for Data Visualization*). SQLite Database Server is used for storing and managing data in the cloud.

*Application layer* contains software application for monitoring data processed in the cloud and presented to system users. The application is light web application with responsive design that can be adjusted to different user devices (desktop computer, mobile phone, tablet).

#### 4. Conclusion

A short introduction into fundamental principles of smart manufacturing is presented, followed with the discussion of development and implementation of specific software architecture in smart

industrial settings. The authors experience with the layered sensor-based software architecture for smart manufacturing systems is presented. The main contributions of the paper are: (1) analysis of software architecture styles in smart manufacturing systems based on the literature review, (2) identification of challenges in implementing software architecture styles in smart manufacturing systems based on the literature review, and (3) presented experience with the layered sensor-based architecture.

Future work includes development of a service oriented architecture model for wireless sensor-based industrial settings, as well as a method for migrating traditional and old fashioned industrial settings to service oriented smart manufacturing settings.

## References

- [1] Kagermann H, Wahlster W and Helbig J 2013 Securing the future of german manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group. National Academy of Science and Engineering. Frankfurt, Germany
- [2] Bassi L 2017 *Proceedings of the IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)* pp 1–6
- [3] Xu L D, Xu E L and Li L 2018 *International Journal of Production Research* **56**(8) 2941–2962
- [4] The Industrial Internet Consortium 2019 *The Industrial Internet of Things, Volume G1: Reference Architecture, Version 1.9* (Milford, MA, USA)
- [5] Fraile F, Sanchis R, Poler R and Ortiz A 2019 *Applied Sciences* **9** 4433
- [6] Li Q, Tang Q, Chan I, Wei H, Pu Y, Jiang H, Li J and Zhou J 2018 *Computers in Industry* **101** 91–106
- [7] Kavakli E, Buenabad-Chavez J, Tountopoulos V, Loucopoulos P and Sakellariou R 2018 *Proceedings of the 6th International Conference on Enterprise Systems (ES)* pp 36–43
- [8] Li F, Fröhlich J, Schall D, Lachenmayr M, Stückjürgen C, Meixner S and Buschmann F 2018 *Proceedings of the 23rd European Conference on Pattern Languages of Programs EuroPLoP '18* (Irsee, Germany) pp 1–11
- [9] Richards M 2015 *Software Architecture Patterns: Understanding Common Architecture Patterns and When to Use Them* 1st ed (Sebastopol, CA, USA: O' Reilly Media)
- [10] Ingenu J 2018 *Software Architect's Handbook* (Birmingham, UK: Packt Publishing)
- [11] Vater J, Harscheidt L and Knoll A 2019 *Proceedings of the 28th International Conference on Computer Communication and Networks (ICCCN)* (Valencia, Spain) pp 1–7
- [12] Theorin A, Bengtsson K, Provost J, Lieder M, Johnsson C, Lundholm T and Lennartson B 2017 *International Journal of Production Research* **55**(5) 1297–1311
- [13] Li S, Xu L, Wang X and Wang J 2012 *Enterprise Information Systems* **6**(2) 165–187
- [14] Cai H, Xu L D, Xu B, Xie C, Qin S and Jiang L 2014 *IEEE Transactions on Industrial Informatics* **10**(2) 1558–1567 ISSN 1941-0050
- [15] Papazoglou M P, van den Heuvel W J and Mascolo J E 2015 *IEEE Software* **32**(3) 61–69
- [16] Chaplin J, Bakker O, de Silva L, Sanderson D, Kelly E, Logan B and Ratchev S 2015 *IFAC-PapersOnLine* **48** 2065–2070 15th IFAC Symposium on Information Control Problems in Manufacturing
- [17] Wan J, Tang S, Shu Z, Li D, Wang S, Imran M and Vasilakos A V 2016 *IEEE Sensors Journal* **16**(20) 7373–7380
- [18] Lucas-Estan M C, Raptis T P, Sepulcre M, Passarella A, Regueiro C and Lazaro O 2018 *Proceedings of the 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)* (Isola, France) pp 37–44
- [19] Bigheti J A, Fernandes M M and Godoy E P 2019 *Proceedings of the 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)* (Naples, Italy) pp 438–443
- [20] Pontarolli R P, Bigheti J A, Fernandes M M, Domingues F O, Risso S L and Godoy E P 2020 *Proceedings of 2020 IEEE International Workshop on Metrology for Industry 4.0 IoT* (Roma, Italy) pp 245–249
- [21] Cao H, Yang X and Deng R 2021 *IEEE Transactions on Automation Science and Engineering* **18** 205–215
- [22] Petrov N, Dobrilović D, Mirosavljev N and Grujić N 2016 *Proceedings of the 8th scientific-professional conference Information technology for e-education* (Banja Luka, Bosnia and Hercegovina) pp 170–176
- [23] Dobrilović D, Malić M and Malić D 2019 *Interdisciplinary Description of Complex Systems* **17**(3-A) 430–437