Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 99 (2021) 598–603

www.elsevier.com/locate/procedia

14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME '20

Cyber-physical systems (CPS) in supply chain management: from foundations to practical implementation

Flavio Tonelli[a,*], Melissa Demartini[a], Massimo Pacella[b], Roberta Lala[c]

[a]Dept. of Mechanical Engineering, Energetics, Management and Transportation, University of Genoa, Via alla Opera Pia 15, Genoa 16100, Italy
[b]Department of Engineering for Innovation, University of Salento, Centro Ecotekne Pal. O - S.P. 6, Lecce 73100, taly
[c]Vigili del Fuoco di Lecce, Italy

* Corresponding author. Tel.: (+39) 010 33 52888; fax: +0-000-000-0000. *E-mail address:* flavio.tonelli@unige.it

Abstract

Since 2015 developments such as Industry 4.0 and cyber-physical production systems on the technology side, and approaches such as flexible and smart manufacturing systems hold great potential. These in turn give rise to special requirements that the production planning, control and monitoring, among others, needing a paradigm shift to exploit the full potential of these methods and techniques. Starting from foundations in Cyber Physical Systems (CPS), building upon definitions and findings reported by literature, a practical example of innovative Cyber Physical Supply Chain Planning System (CPS^2) is provided. The paper clarifies the advantages of cyber-physical systems in the production planning, controlling and monitoring perspective with respect to manufacturing, logistics and related planning practices. A set of basic features of CPS^2 systems are discussed and addressed by contextualizing service orientation architecture and microservices components with respect to supply chain management collaboration and cooperation practices. The identification of specific technologies behind those functions, within the developed research, provides some practical insight if the interesting CPS^2 potential.

Keywords: Cyber-physical systems; Supply chain Management; Production planning and control; Service-oriented architecture; Smart manufacturing

1. Introduction

Information and communication technologies (ICT) have been growing very fast in the last few years, helping companies to manage and speed up production and organizational processes as well as to increase supply chain's (SC) visibility, increasing the value of the manufacturing activities [1]. These technologies can enable real-time response to manage demands and/or conditions in the factory, in the SC network and in customer needs [2]. In this regard, the implementation of cyber physical systems (CPS) can lead to additional benefits. CPS are based on the integrations of digital and physical processes through a digital twin (DT) [3, 4]. CPS include embedded systems such as equipment, buildings, transportation means, and devices, but also logistic, coordination and management processes as well as internet services. CPS collect, manage and analyze data through the support of sensors, while actuators are used to react to production or organizational changes and communicate with the other components. CPS can be implemented in order to manage different issues such as production, logistics, quality, planning and scheduling activities within the factory or the SC [2]. Therefore, the scope of this paper is to provide a practical example of innovative Cyber Physical Supply Chain Planning System (CPS^2) with respect to new approach in manufacturing, logistics and related planning practices. A set of basic functions of CPS^2 systems are discussed and addressed. Then a brief explanation of service orientated architectures and microservices, as well as their integration is provided. The identification of specific technologies behind those architectural aspects are then presented and discussed to explain the great potential of a CPS^2. The paper is organized as follows: in Section 2 a qualitative literature review on CPS is provided, Section 3 distinguishes between service-oriented architecture (SOA) and

10.1016/j.procir.2021.03.080

microservices, section 4 presents the integration of these services by enterprise service bus (ESB) and section 5 describes the developed CPS^2 architecture and main features. In section 6 the system is disccused and finally in section 7 conclusions are summarized.

2. Qualitative literature review

CPSs interact and manage the physical system through networks, containing a large number of physical systems and components such as machines, conveyor belts, robots, wireless sensors net 'to enable' the digital representation of the 'production world'. CPSs are based on a smart network to collect and manage data in order to make decisions on the physical world by using a digital representation of that, namely the digital twin (DT). This capability gives CPS a hierarchical view as data and information can be managed through different production levels from the operational up to the strategic one. CPSs transfer data to manage operations, guide users and to add resilience to the manufacturing system through a 'real world' evidence-based decision making.

According to [5], CPS it is a key technology for realizing smart factory, and it is being studied in close relationship with such technologies as:

- *Plug and produce*: A plug and produce system can be described as a collection of stations or modules for assembling or checking parts. Modules can be replaced with others having similar functionality and interfaces in case of breakdown or to adapt to a new process. Also, new modules can be added to increase production volumes [6]. The concept follows a product-centric approach, in fact the product is the driver of its own production, there is no need for central coordination. Production systems are composed of intelligent production units that are able to configure themselves, execute a defined set of production skills autonomously, or in cooperation with other units [7]. A production unit is aware of its production skills, capabilities, state, and its physical and virtual environment. Different production units could be identified such as machines, robots and conveyors.
- *Smart products*: Smart products are products that are capable to do computations, store data, communicate and interact with their environment. To this end, it is necessary to develop chips and microprocessors as well as embedded systems [8]. Smart products describe their properties, status and history. They are able to communicate and exchange data and information about their lifecycle. The capability to individually specify its properties can be used for an individual production with varying size. Smart products interact with their physical environment. Sensors allow to capture physical measures, cameras to get visual information on the product and its environment in real-time. Actors enable products to impact physical entities in their environment without human intervention.

CPS represent the next generation of innovative systems [9] integrating communication, computing and control to increase stability, performance, reliability, robustness, and efficiency. Furthermore [10] argues that CPS is the integration and collaboration between hardware and software in order to build a smart factory which aims at enabling efficient end-to-end workflows and new forms of user-machine interaction, in a wide range of application fields, while [11] underline that CPS are complex and multi-disciplinary engineered systems that integrate computing technology into the physical phenomena by using transformative research approaches. This integration mainly includes observation, communication, and control aspects of the physical systems from the multi-disciplinary perspective. Finally, [12] summarize the key characteristics of a CPS (Table 1).

CPS are often used in connection with microservices. CPS microservices are described using web technologies and are available for discovery and use during the development time of the manufacturing system processes, but also during its operation to have a flexible manufacturing system able to address the challenge of mass customization [13]. [14] highlight that a special focus should be put on the shift of CPS towards service-oriented architectures. Two aspects that may be hard to combine in a single 'ecosystem' design having to consider when a service-oriented architecture (or microservices) is adopted to connect conceptual models and CPSs.

Table 1. Overview of the main CPS characteristics

Characteristics of CPS	Description
Heterogeneity	CPS integrate several different systems together with standard communication and information exchange. They integrate various devices, including sensors, mobile devices, workstations and servers.
Interoperability	It is the capability of system components to connect, communicate, and operate with each other. Interoperability allows CPSs to exchange mutually intelligible information.
Interconnection	CPS are composed of processing elements and physical elements in large-scale wired and wireless networks through a variety of sensors and actuators, aiming at constructing intelligence across different fields.
Modularity	CPS are modularized, flexibly changed, and reconfigured in response to rapidly changing customer needs and product changes. Modularity allows system independence, making it capable to adapt more flexibility.
Autonomy	CPS are able to independently learn and adapt to the environment. Autonomy brings to the self-capabilities of CPSs. Self-capabilities can in fact be seen as exemplifications of autonomy. Instances of self-capabilities are self-adaptivity, self-reconfiguration, self-organization, self-awareness, self-learning, self-diagnosis, self-healing, self-optimization, self-protection, and self-explaining.
Decentralization	CPS work independently and make decisions autonomously in a way they remain aligned with the path toward the single ultimate organizational goal.
Integration	CPS are integration of computation and physical processes. Many authors described virtualization as the ability to link sensor data to virtual factory models and simulation models; in other words, virtualization consists in creating a virtual copy of the real physical world and remaining connected to it overtime.

This because of the high variability of conceptual models at design time and the complex behavior of CPSs at run-time. For the latter, service-oriented architectures provide a structure in which the commitment to abstract, intricate functional capabilities of CPSs can be realized.

3. From service-oriented architecture to micro-services

The traditional approach to supply chain management applications is based on monolithic applications requiring that all components and functionalities, that can be deployed, are contained within the same application. However, this method has disadvantages: the larger the application, the more difficult it will be to quickly solve new problems and add new features. Both service-oriented architectures and microservices simplify troubleshooting and optimize development and response in evolving specific domain solutions. Differently from SOA, microservices are both an architecture and an approach to writing software. With microservices, applications are broken down into their smallest elements, independent of each other. Compared to the traditional monolithic approach, whereby each component is created within a single element, the microservices interact to complete the same activities, remaining independent of each other. Each component, or process, represents a microservice; this type of software development approach promotes granularity and allows you to share similar processes among multiple apps by optimizing application development, allowing to approach a native cloud model.

Each function, namely a microservice, can be compiled and implemented independently. As stated by [15], a microservice architecture is an approach to the development of a single application as a set of small services, each running in its own process and communicating with light mechanisms. Designed and developed around the domain's business capabilities, these services can be deployed independently and in a fully automated way. The logic of these services can be implemented with different programming languages and use different storage technologies. Therefore, individual services can work, or not, without compromising others.

The architecture based on microservices not only involves the low coupling between the basic functions of an app but proposes a radically different concept of the development teams and of the communication framework between the interacting services. This approach offers the possibility to manage unavoidable critical issues, supports dynamic scalability and facilitates the integration of new features as the ones emerging in new dynamic evolving paradigm of next generation supply-chain. To deploy microservices and take advantage of this approach, basic elements of a SOA need to be adopted and adapted. Within the SOA architecture, the apps are structured in reusable services that communicate with each other via an Enterprise Service Bus (ESB), also called Event Bus. Microservices, on the other hand, can communicate with each other, generally in stateless mode, allowing to build apps with greater fault tolerance and less dependent on a single Enterprise Service Bus (ESB).

Figure 1 show main differences between the two approaches.

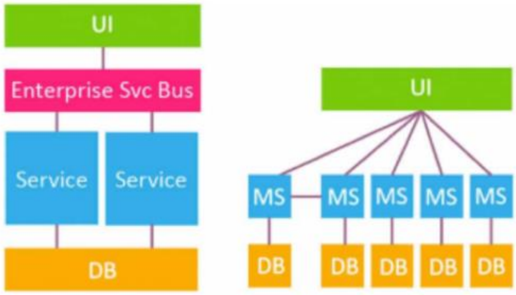

Fig. 1. SOA vs Microservices Structure.

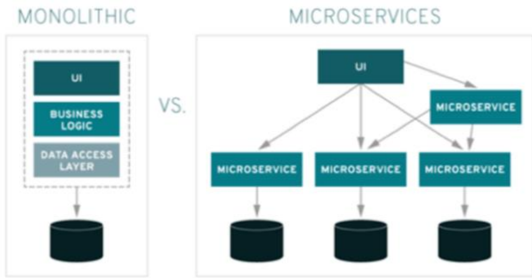

Fig. 2. Monolithic vs Microservices Structure.

In addition, they communicate through language-independent application programming interfaces (APIs) and this allows development teams to choose their own tools.

Considering the evolution of SOA, microservices are not an absolute novelty, but lately they have become more attractive thanks to the advances in containerization technologies. In practice, microservice architecture can be considered an evolution of SOA architecture.

In figure 2 is represented a diagram that compares the monolithic approach with microservices.

Micro-services advantages

Based on a distributed architecture, microservices allow for more efficient development and routines. The ability to develop multiple microservices simultaneously allows multiple developers to work on the same app simultaneously, reducing development time.

- **Time to market:** by allowing you to shorten development cycles, a microservice-based architecture supports more agile deployments and updates.
- **Scalability:** as demand for certain services increases, microservices can be distributed across multiple servers and infrastructures, based on business needs.
- **Resilience:** each service, if built correctly, is independent and does not affect the other services in the infrastructure. Consequently, the possible error of a component does not block the entire app, as happens with the monolithic model.
- **Deployment:** since microservice-based apps are smaller and more modular than traditional monolithic applications, all problems associated with such deployments are automatically eliminated. Although this approach requires superior coordination, the resulting benefits are crucial.

- **Accessibility:** since the larger apps are divided into smaller parts, it is much easier for developers to understand, update and improve these components.
- **Openness:** thanks to the language-independent APIs, developers are free to choose the optimal language and technology for the function to be created.

Micro-services disadvantages

An architecture based on microservices presents two main critical issues namely complexity and productivity [16].

- **Compilation:** time must be devoted to identifying dependencies between services and data. Because of these dependencies, many others may need to be run when you compile.
- **Tests:** integration tests, are more important because an error in one part of the architecture could cause an error in a component at various steps, depending on how the services are structured to support each other.
- **Versioning:** upgrading to a new version may compromise backward compatibility, solved by using conditional logic, with multiple live versions for different clients and more complex maintenance and management.
- **Deployment:** during initial configuration, investing heavily in semi-automated solutions since complexity of microservices would make manual deployment extremely difficult.
- **Registration:** in distributed systems, centralized registers are needed to reconnect all the various components, without which managing them in a scalable way would be impossible.
- **Monitoring:** it is essential for the operating teams to have a centralized visibility of the system, in order to identify the origin of the problems.
- **Debugging:** remote debugging is not a viable choice with dozens or hundreds of services. There is currently no single solution related to debugging.
- **Connectivity:** it is necessary to evaluate which method of detection of the services to choose, whether centralized or integrated.

4. Micro-services integration through ESB: the backbone for supply chain planning and management

One of the founding pillars of a CPS devoted to supply chain planning and management is the communication and collaboration capability among the different actors composing the chain. The events generated by the various actions performed by these actors must be propagated both inside and outside the system boundaries, on the one hand to interact with other systems (management, accounting, analysis, etc.), on the other to get up to user of the single web app who must be notified of an event that could alter his/her work flow. These two aspects are enslaved by two distinct subsystems: one is an event communicator, the other is a notification system for web apps. This can be done by implementing an Enterprise Service Bus (ESB), figure 3, providing for the subdivision into technological and/or application 'islands', through the following functionalities:

Fig. 3. Enterprise Service Bus Structure.

- **Routing:** provides the infrastructure with the ability to sort a request to a particular service provider using deterministic or probabilistic criteria that is depending on the decisional process of supply-chain.
- **Transformation:** converts the structure and format of the payload of the request made by the client into the format effectively managed by the service provider allowing different actors 'playing the same game' even by working on separate, diverse enterprise information or legacy systems.
- **Message Enhancement:** can add, modify or delete information contained in a message in order to make it compatible with the service provider. For example, with this feature the bus can convert the date format or add information not originally present allowing for international implementation in different languages, format, time zones, …
- **Message Processing:** manages the status and requests ensuring the delivery of the reply message to the client supporting the coordination of preset workflow as well as asynchronous emergent requests.
- **Protocol Transformation:** accepts a type of protocol towards the client (i.e. SOAP, JMS) and communicates to the service provider through another protocol (eg RMI). Protocol Transformation is used to send the same payload using a different protocol.
- **Message Transformation:** changes the format and values of the payload that travels between client and service provider in order to facilitate the interoperability.
- **Service orchestration:** acting as a centralized coordinator (broker) who controls the services involved and coordinates the execution of the different operations. For this functionality, it is used as standard BPEL (Business Process Execution Language) language, but there is the possibility of using other languages such as Business Process Modeling Notation (BPMN) or Web Service Conversation Language (WSCL) that can be compatible with workflow tools.
- **Transaction Management:** treats a request to a business service as if it were a single unit of work allowing for a synchronization between transactional system (ERP) and planning system (SCM).
- **Security:** gives the infrastructure the ability to protect services from unauthorized access. The following are therefore essential: authentication, authorization, auditing and administration protecting technological know-how and other relevant business information.

The ESB chosen for the proposed CPS^2 is RabbitMQ which implements event-based communication between different microservices developed using ASP.NET Core. RabbitMQ is a widely used open source message broker.

5. From supply chain planning to cyber-physical supply chain systems: CPS^2 for next generation supply planning

The proposed CPS^2 is composed of 3 levels: front-end (local client), APP and algorithms (cloud server), database and integration (local server).

The development of the front end contains a dynamic data representation engine, common to all applications, strictly aimed at an analysis and real-time forcing of the supply chain planning simulation data. Starting from the bottom, different data sources can be connected: ERP, PLM, CRM in order to collect, transform, and store these real-world data into the CPS^2 database management system. These data can also be provided by Internet of Things connectors (this feature represents the real foundation for the cyber-physical system definition, which communicates directly with the edge computing layer (data collection from machines). All the collected data are historicized within a Big Data Service (BDS) and made available to the whole company.

The Enterprise Service Bus (RabbitMQ) is the heart of the CPS^2: the algorithmic engine is then composed (for example) by algorithms to perform forecast [17], by the algorithms for planning, by the algorithms for scheduling, and so on. These algorithms are used by the CPS^2 Smart Services and then executed in the CPS^2 framework.

To overcome statistical complexities in analyzing time series, in [17] a deep learning system has been described for demand forecasting. Given that nonlinear and non-stationary dynamics pose the major challenge for accurate demand forecasting in supply chain management, the problem has been approached by implementing the long short-term memory network. The results in [17] indicate that the proposed deep neural network is a competitive method and is able to outperform state-of-the-art statistical forecasting methods in supply chain management, where the future demand for a certain product is the basis for the respective replenishment systems.

The Simulation Environment allows of having a number of simulations active for each user, based on a virtual representation of the different supply-chain components, each with their own display layout, on which users can cooperate and interact by using the Cooperative Layer.

In addition, it is possible to perform the engagement between two users on a simulation, whereby two users share the same simulation data and the changes made by one are simultaneously received by the other as well as coordinating the flow of activities between users in order to accomplish specific tasks according to a specific workflow as configured.

The proposed CPS^2 main components, allowing for a complete management of digital supply chain planning, execution, and control are listed:

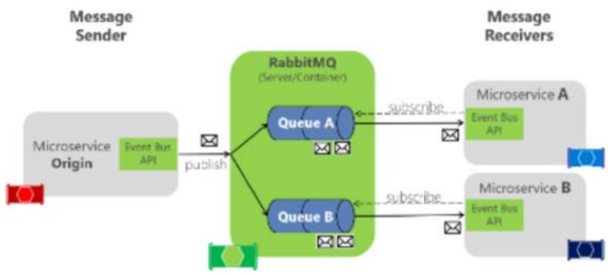

Fig. 4. ESB implemented in RabbitMQ.

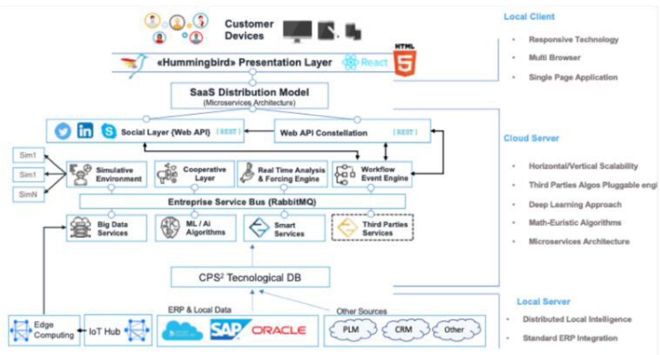

Fig. 5. CPS^2 General Architecture.

- **Data Base:** structured on two separate logical levels (Static Layer and Simulative Layer) that communicate through structured events and functions and guarantee the persistence and consistency of the data entering the system and the fluidity of the simulative cyber-physical representation.
- **Business Intelligence:** analytics application able to represent graphically even large sets of data using both third-party application and internal development. Big Data is the engine for storing and analyzing large volumes of data and mathematically analyzing of historical data at different levels of aggregation.
- **Workflow Manager:** an engine capable of performing functions of the CPS^2 universe and/or invoking utilities in the Microsoft Azure framework. The Workflow is a subscriber of the Event Management system and can be invoked upon the occurrence of any type of event.
- **Artificial Intelligence:** a set of engines and functions that allow the use of machine learning and deep learning logics to empower the CPS^2 data analysis and interpretation as well as semi-autonomous and self-learning decision support. Recently, deep learning has received a well-deserved research attention in the literature and a few studies have been presented concerning the use of deep neural network as autonomous support system for supply chain management, in particular with reference to demand forecasting [17].
- **Event Manager:** core service for receiving, processing and dispatching messages relating to events triggered by the system (ESB) accordingly to Microsoft Flow engine in order to empower coordination and cooperation between supply-chain actors.

6. Discussion

With respect to the overview of the main CPS characteristics as reported by table 1 the proposed CPS^2 guarantees:

- **Heterogeneity:** supply chains consist of parts or things that are very different from each other and the proposed CPS^2 is capable to model and represent all these components through the adoption of interconnected microservices.
- **Interoperability:** the degree to which two components, services, programs, etc… can be used together, or the quality of being able to be used together is guarantee by the general CPS2 architecture as shown in figure 5.
- **Interconnection:** the connection with other things that are related to each other, as typically occurring in supply chains, is supported by the ESB implementation as well as by network early- and late-binding of components, during the execution phase.
- **Modularity:** CPS^2 services can become increasingly modularized creating a new dynamic infrastructure representing the dynamic change of the modelled supply-chain network.
- **Autonomy:** much of the sensing, estimation, and fusion of data that enables applications of autonomous cyber-physical systems increasingly rely on deep learning and similar machine learning techniques enabled by recent advances in artificial intelligence through approaches such as deep neural networks, embedded in so-called learning enabled components that accomplish tasks from classification to control and forecasting.
- **Decentralization:** the CPS^2 allows to decentralize the governance of supply-chain by moving, partially, the control from a single place (i.e. supply-chain control tower) to several smaller ones corresponding to services.
- **Integration:** the proposed CPS^2 approach combines these virtual services, data, systems, and cognitive capabilities in order to be more effective in providing decentralized, self-adapting and autonomous abilities.

7. Conclusions

Cyber Physical Production Systems (CPPS) are considered as the next evolution for the design of production and transport processes in supply chains. The underlying concept aims to enhance the overall performance of distributed and autonomous processes that collaborate in networks. The framework has the cyber-physical microservice as a key construct for the modeling of the system. Microservices and the SOA approach in the CPS have the same goal: building one or multiple applications from a set of different services. However, in the microservices approach companies have made considerations how these individual distributed services need to be designed to work together properly. The CPS^2 model described in this paper aims at collecting and analyze data in real time to plan simulation data.

Acknowledgements

This work has been funded by Puglia Region (Italy) – Project 'Cooperative Supply Chain'.

References

[1] Demartini M, Orlandi I, Tonelli F, Anguita D. Investigating sustainability as a performance dimension of a novel Manufacturing Value Modeling Methodology (MVMM): from sustainability business drivers to relevant metrics and performance indicators. Proceedings of the Summer School Francesco Turco, 13-15 September 2016, pp. 262-270.

[2] Klötzer C, & Pflaum A. Cyber-physical systems as the technical foundation for problem solutions in manufacturing, logistics and supply chain management. 5th International Conference on the Internet of Things (IOT), IEEE 2015:12-19.

[3] Revetria R, Tonelli F, Damiani L, Demartini M, Bisio F, Peruzzo N. A real-time mechanical structures monitoring system based on digital twin, iot and augmented reality. Spring Simulation Conference, SpringSim 2019, art. no. 8732917.

[4] Damiani L, Demartini M, Giribone P, Maggiani M, Revetria R, Tonelli F. Simulation and digital twin based design of a production line: A case study Lecture Notes in Engineering and Computer Science 2018:2.

[5] Demartini M, Tonelli F, Damiani L, Revetria R, Cassettari L. Digitalization of manufacturing execution systems: The core technology for realizing future smart factories. Proceedings of the Summer School Francesco Turco, September 2017, pp. 326-333.

[6] Weyer S, Schmitt M, Ohmer M, Gorecky D. Towards Industry 4.0-Standardization as the crucial challenge for highly modular, multi-vendor production systems. Ifac-Papersonline 2015;48(3):579-584.

[7] Jatzkowski J, Adelt P, Rettberg A. Hierarchical Scheduling for Plug-and-Produce. Procedia Technology 2016;26:227-234.

[8] Abramowicz W. Industry 4.0 Potentials for creating smart products: empirical research results. Lecture Notes in Business Information Processing 2015.doi:1007/978-3-319-19027-3.

[9] Kim KD, Kumar PR An overview and some challenges in cyber-physical systems. J of the Indian Inst of Science2013;93(3):341-352.

[10] Begh, A, Marcuzzi F, Rampazzo M, Virgulin M. Enhancing the simulation-centric design of Cyber-Physical and Multi-Physics Systems through co-simulation. In 17th Euromicro Conf. on Digital System Design (DSD) IEEE 2014, pp. 687–690.

[11] GunesnV, Peter S, Givargis T, Vahid F. A survey on concepts, applications, and challenges in cyber-physical systems. KSII Transactions on Internet and Information Systems 2014;8(12):4242–4268.

[12] Napoleone A, Macchi M, Pozzetti A. A review on the characteristics of cyber-physical systems for the future smart factories. J of Manuf Syst 2020;54:305-335.

[13] Thramboulidis K, Vachtsevanou DC, Solanos A. Cyber-physical microservices: An IoT-based framework for manufacturing systems. Industrial Cyber-Physical Systems (ICPS) IEEE. 2018: 232-239.

[14] Walch M. Operating cyber-physical systems with microservices: the s* IoT conceptual modelling approach. 7th International Congress on Advanced Applied Informatics (IIAI-AAI) IEEE 2018: 787-792.

[15] Lewis J, Fowler M. Microservices. MartinFowler. com. 2014

[16] Sampaio AR, Rubin J, Beschastnikh I, Rosa NS. Improving microservice-based applications with runtime placement adaptation. J of Internet Services and Applications 2019;10(1).1-30.

[17] Pacella M, Papadia G. Evaluation of deep learning with long short-term memory networks for time series forecasting in supply chain management. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering 2020.