Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 72 (2018) 486–491

www.elsevier.com/locate/procedia

51st CIRP Conference on Manufacturing Systems

Modular System Design Approach for Cyber Physical Production Systems

Emmanuel Francalanza[a,]*, Mark Mercieca[a], Alec Fenech[a]

[a]CERU, Department of Industrial and Manufacturing Engineering, University of Malta, MSD 2080, Malta

* Corresponding author. Tel.: +356 2340 206; *E-mail address:* emmanuel.francalanza@um.edu.mt

Abstract

As manufacturing enters a new paradigm of cyber production, new methods are required to support the production system design activity. These methods have to take into consideration both the cyber and physical aspects of the system design, whilst also satisfying requirements of modularity, connectivity and intelligence. This paper presents a modular system design approach for cyber-physical production systems, which is based on the established systematic method of modular function deployment. The result of applying this design approach is a modular system architecture which describes the system modules for both cyber and physical aspects of the production system.

Keywords: Industry 4.0; Design Process; Production Systems;

1. Introduction

Cyber-Physical Production Systems (CPPS) consist of autonomous and cooperative elements (e.g. Smart Machines) and sub-systems (e.g. Smart Factories). These elements are connected and can communicate with each other in situation dependent ways, on and across all levels of production, from field device and processes level, up to the factory and production planning levels [1]. The “Internet of Things (IoT)” and “Cyber-Physical Systems (CPS)” [2] are terms that have been coined to describe these engineered and connected systems, which have embedded computational and networking capabilities. The full industrial implementation of CPPS in manufacturing environments is bringing about the fourth industrial revolution or as defined by the German Federal Minister of Education and Research, Industry 4.0 [3].

1.1. The need for CPPS

There are various drivers for the development and implementation of CPPS. These drivers may vary significantly, such as social and political factors, which are forcing new legislation and concepts of sustainable development. A central driver for the implementation of CPS are new and cutting edge developments driven by connectivity. This new reality is being brought about by advances in technologies such as low power electronics and wireless communications. From a manufacturing perspective, this technological drive is brought on by the development of technologies such as smart sensors [4], cloud manufacturing [5], advanced communication protocols such as OPC/UA [6] and data models such as AutomationML [7].

As explained by ElMaraghy [8] et al. and Koren [9] this is bringing about an evolution in manufacturing. This evolution is transforming manufacturing from the Taylor paradigm in the 1920s, to a new paradigm of personalised products and production. Due to this shift in customer needs, together with highly competitive markets and faster technological advances, products are constantly changing and evolving over time. ElMaraghy [10] states that product ranges continuously evolve with the addition of new features or parts that may be added or replaced to the current range of products.

Together with an increase in product variety, and an associated decrease in production volume per variant, this new paradigm has also resulted in a reduction of the product life cycle. This means that the factories and production systems (PS) that have been planned to manufacture these products have longer life cycles than the products which they produce. Hence, as argued by Schenk et al. [11], the inherent nature of

10.1016/j.procir.2018.03.090

factories means that they need to be capable of changing their production capabilities in order to produce different and evolved product ranges throughout their life time.

1.2. Challenges in designing CPPS

Chryssolouris [3] defines PS design as the mapping from performance requirements of the PS onto suitable values of decision variables, which describe the physical design or the manner of operation of the production system. PSs can be modelled as complex [12] technical systems from a systems theory perspective. This is the basis for the paradigm “a factory is a long life and complex product” [13]. Therefore, as explained by Francalanza [14], PSs need also to be designed using systematic approaches.

Typical PS design problems include manufacturing resource requirements, resource layout, material flow and buffer capacity [15]. In these type of PS design problems detailed requirements and constraints are well established [16]. Solutions are developed and then evaluated based on the requirements. These types of resource optimization and traditional factory layout problems are considered to be “tame” or “benign” problems [17], since the objectives are clear, and it is in turn clear whether or not the problem has been solved. As argued in the previous section, products are constantly evolving during the life time of the PS. We can therefore argue that PS requirements are changing with time. Therefore the objectives of a PS are often not well defined during the early PS design process, and are subject to change [16].

In contrast to these “tame” type of problems, Francalanza [18] refers to the PS design problem as “wicked problems”. This is based on Rittel and Webber’s, definition for wicked problems [17]. The primary difficulty with wicked problems is that their requirements are ill-defined. Due to this uncertainty in defining the problem, it is equally difficult to determine if a solution has been found. This is comparable to a Class III problem as defined by Ueda et al. [19], i.e. a problem with incomplete specification, where not only the environment description but also the specification is incomplete.

1.3. Designing modular CPPS

One approach for handling wicked PS design problems is to design modular CPPS which are capable of changing and evolving their capabilities based on their changing requirements [14]. By designing and implementing modular CPPS the production capability may evolve by adding or replacing elements, such as by plugging in new machines, robots, and material handling systems.

A system architecture which defines the system elements and their interface is therefore required in order to design and implement a modular CPPS. As will be discussed in Section **Error! Reference source not found.**, existing literature does not provide an approach which supports the development of such a system architecture. This need therefore forms the basis for the motivation of this research work. The research aim was to develop a system architecture developed by using a modular system design approach, which is presented in Section 3. Section 4 presents a modular CPPS based on this system architecture.

2. Approaches to CPPS Design

The concept of developing modular PSs is not new, and was described by Rogers and Bottaci in [20]. An approach for implementing such modular and changeable PSs is contributed by Schuh et al. contribute which is based on an object-oriented design method [21]. From a CPS perspective an interesting approach to design is the one contributed by Nuzzo et al. [22] who developed a platform-based design method for the design of CPS. In this approach Nuzzo uses design by contract, a software design approach, to specify and abstract the components of a CPS, and provide formal support to the entire design flow. Another approach to support CPPS design is the use of reference models. One such model is the Reference Architectural Model Industrie 4.0 - RAMI 4.0 [23]. This model consists of a three-dimensional coordinate system that describes all crucial aspects of Industrie 4.0. In this way, complex interrelations can be broken down into smaller and simpler clusters providing a common understanding of the relations between individual components of Industrie 4.0 solutions. A similar model to RAMI 4.0 is the Smart Grid Architecture Model - SGAM [24], which can act as a reference designation system in order to describe smart grid (technical) use cases as well as business cases. In [25], Stark et al. present an architecture design approach for modularized design of CPPS. This systematic and model based design approach for flexible PSs provides the ability for interdisciplinary concurrent engineering, validation and testing. The need for continuous adaptation has also driven the development of approaches that implement the concept of plug-and-produce. Plug-and-produce allows for different elements of a PS to be added and removed from the PS depending on the needs of production. In fact as argued by Onori [26], evolvable systems are not only adaptable to change but assist in the evolution of their elements in time, such that processes may become self-evolvable, self-reconfigurable, self-tuning and self-diagnosing.

As explained by several authors such as Schleipen et al [27], Onori [28] and Maeda [29], the concept of plug-and-produce must be supported not only from a mechanical function, but also by the development of new and improved software and control paradigms. In fact to support the plug-and-produce concept there needs to be the development of system models that accurately represent the PS in all its domains, mechanical, electrical and cyber-space. One such approach is the Line Information System Architecture (LISA) which provides the possibility of integrating devices and services on all levels, simplifying hardware changes and integration of new smart services as well as supporting continuous improvements on information visualisation and control [30]. The concept of self-awareness is an important requirement of modular PS and has been successfully implemented through the service-oriented architectures (SOA) in the “Factory of Things” [31]. Agent based design concepts such as holonic PSs [32] have similarly shown promise in this area.

Whilst system architectures and CPPS design approaches do exist in literature, none of the above mentioned studies provide a systematic design approach for modular CPPS. This study therefore addresses this research gap by contributing a modular CPPS design approach as described in the following section.

3. Modular CPPS Design Approach

As described by Wiendahl et al. [33] one of the enablers of changeability is modularity. Modularity provides the possibility of adding or removing modules to existing systems, hence allowing for changes in the system capabilities, both in terms of the product feature and production volume which can be produced. Modularity is a concept which was first developed within product design, and allowed for mass customization through the development of product families and product platforms [34]. The system architecture defines the modules and their interfaces [35]. A system architecture which defines the modules and their interfaces is required to develop a modular CPPS. This Section therefore presents and contributes a design approach which can be utilized to develop system architectures for modular CPPS.

3.1. Modular Function Deployment Method

The systematic approach employed by this research is based on the Modular Function Deployment (MFD) [36]. The MFD method was developed by Erixon [36] to support product designers in developing modular products by identifying which parts could be combined into modules, based on a set of module drivers. As described in Section 1.2, since from a systems theory perspective a CPPS can be considered as a product, this product design method is here being applied to develop modular CPPS. The MFD method which is being used for this research is illustrated in Fig. 1. This method consists of a number of steps. As illustrated in this figure different steps make use of systematic methods to support the designers in developing a modular CPPS architecture.

3.2. Step 1 – Clarify Requirements

The first step in this method is to clarify the system requirements and transform them into design specifications. In designing CPPS, this method combines the typical requirements of the PS, such as safety and production process, to the specific requirements of CPPS, such as intelligence, security and networkability.

To support this activity the Quality Function Deployment (QFD) method is used here to clarify the requirements and formulate the CPPS design specifications. This method uses a modified QFD where each of the CPPS design specifications is rated against the requirements in having a strong (9), medium (3) or weak (1). The CPPS specifications, i.e. the "how" requirements are achieved are classified as cyber or physical to show the dual nature of CPPS. Secondly modularity for both the Cyber and Physical aspects is listed, to accentuate the importance and main aim of this method.

Fig. 2 represents the QFD for CPPS conducted during this research.

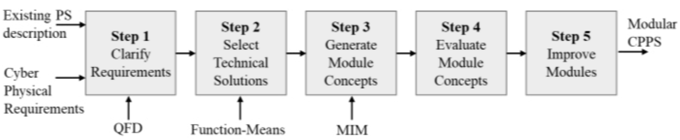

Fig. 1- CPPS Modular Function Deployment

CPPS Requirements ("What")	Cyber: Modularity	Cyber: Function Execution	Cyber: Connectivity	Cyber: Data Analytics	Cyber: Digital Twin	Cyber: Interfacing	Physical: Modularity	Physical: Part transportation	Physical: Part Identification	Physical: Transformation Process	Physical: Part Storage
Product Variety	●	●	⊘		⊘	⊘	●	●	●	●	⊘
Adaptive/Changeable	●	●	●	⊘	○		●	⊘	⊘	⊘	⊘
Smart/Intelligent		⊘	⊘	●	●	⊘			⊘	⊘	
Production Process	⊘	●		○	○	○	⊘		○	●	
Safe	○	⊘	○					⊘		⊘	
Secure		⊘	⊘	⊘	○	○			○		
Real-time	○	⊘	⊘	○	○	○			○		
Networked	●	⊘	●	●	●	●	○		⊘		
	32	42	31	26	24	18	22	15	21	27	6

Design Requirements ("How")

● Strong Relation (9)
⊘ Medium Relation (3)
○ Weak Relation (1)

Fig. 2 – Quality Function Deployment for CPPS

The system's cyber capability of functional execution, i.e. the capability of executing out the required logical functions in order to carry out a process achieves the highest score, followed by modularity. From the physical aspect, the transformation process, e.g. assembly, fabrication, testing process achieves the highest score, and is also followed by modularity being rated as highest. This demonstrates that whilst the main functions of a CPPS are to execute logical instructions in order to carry out a production transformation process, modularity is an important functional requirement and should therefore be integrated into the CPPS design process.

3.3. Step 2 – Select Technical Solutions

In order to identify the different elements which make up a CPPS and hence define the modular architecture, the functions identified in the QFD have to be translated into technical solutions. The second step of this method therefore defines how the required functionalities can be implemented. To support this activity the function-means tree is employed to decompose the CPPS functionality to aid concept generation. There are many functions and respective means which need to be established in order to implement a CPPS. Due to space limitations Fig. 3 only illustrates an excerpt of the function-means method for cyber virtualization.

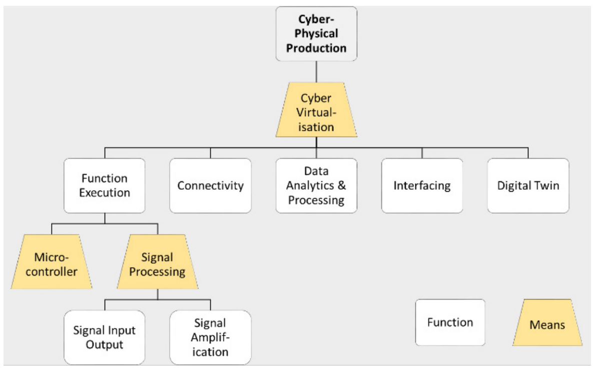

Fig. 3 – Excerpt of function-means for cyber aspects

3.4. Step 3 – Generate Module Concepts

At the heart of the MFE method is the MIM (Modular Indication Matrix) [36]. This matrix, employs a QFD-like approach to provide an indication of which sub-function(s)/elements should be grouped to form a single module. The MIM depicts the relationship between a set of module drivers and the CPPS elements which have been derived from the previous Step 2. The MIM for cyber and physical modularity are illustrated in Fig. 4and Fig. 5 respectively. In the MIM the elements relationship to a set of module drivers is related as being strong (9), medium (3) or weak (1). The elements which have a higher total score, are those elements having a higher relationship with the module drivers, and are to be considered as single modules.

The elements with lower relationship scores may be grouped according to their functionality into modules. The module drivers which are used in this method are based on the drivers developed by Erixon for modular product design [36], but have been modified to the needs of modular CPPS design and are detailed hereunder.

Module Driver		Micro-controller	I/O Device	Motion Driver	Deterministic Protocol	Non-Deterministic Protocol	Mfg. Execution System	Deep Neural Network	Data Model
		Elements							
Develop-ment and Design	Carry-over	○	○	○	⊘	⊘	●	●	●
	Technology	○	○	○	⊘	⊘	⊘	●	●
	Manufacturing Strategy						⊘	⊘	
Variance	Technical Specification	○	⊘	⊘					
	Communication	⊘	⊘	⊘	●	●			⊘
	Safety	⊘	⊘	⊘	⊘	●			
Purchase	Black-box eng	⊘	⊘	⊘	●	●	●	⊘	⊘
Factory Lifetime	Service/maintenance	⊘	⊘	⊘	○	○			
	Upgrading	⊘	○	○	⊘	⊘			
	Reconfiguration	●	●	●					
		27	27	27	40	46	33	33	33

Fig. 4 – MIM for cyber modularity

Module Driver		RFID	Part Routing	Part Location	Part Holding	Part Motion	Pneumatic Valves	Pneumatic Actuator	Position Sensor	Aluminium Framing
		Elements								
Design	Carry-over	●	●	●	⊘	●	⊘	○	○	●
	Technology	○					○	○	○	
	Manufacturing Strategy	○					○	○		
Variance	Technical Specification			●	●	⊘				
	Communication	⊘					⊘			
	Safety									⊘
Purchase	Black-box eng	○					⊘	⊘	⊘	
Factory Lifetime	Service/maintenance	⊘					⊘	⊘	⊘	⊘
	Upgrading	⊘					●			⊘
	Reconfiguration	●		●	●					●
		33	12	30	33	15	32	18	17	33

● Strong Relation (9)
⊘ Medium Relation (3)
○ Weak Relation (1)

Fig. 5 - MIM for physical modularity

The first set of module drivers are related to the development and design activities. These relate to the need for an element to be defined as a separate module because it has to be carried over to new designs, or because it needs to be replaced during PS life cycle due to a technology update. This driver also takes into consideration the need to change an element due to manufacturing strategy, which may be company specific. The second module driver relates to how variation is handled by the CPPS and how it is influenced by variation due to changing CPPS requirements (product feature/ production volume), changing communication protocols and updates in safety requirements. The next driver relates to purchasing of CPPS elements, and explores whether an element since it is being bought-in has to be considered as a separate entity, i.e. as a black-box. The last module driver relates to the factory lifetime and rates the element by rating if it is easier to maintain, upgrade or reconfigure if it is a separate module.

Based on the analysis of the MIM for cyber and physical modularity a number of modularity concepts were developed. For cyber modularity, the results of the MIM indicate that having a low score, the micro-controller, I/O devices and motion drivers can be integrated into a single module. Similarly the manufacturing execution system, deep neural network and data model can also be integrated. For the first concept these were integrated into a single cloud module. In concept two these were left separate with the data model on the cloud, and engineering application such as the manufacturing execution system as isolated modules running on separate machines.

A similar exercise was carried out for physical modularity, were a number of modularity concepts were developed by grouping different elements together. Mainly this involved in grouping the elements into a material handling module, which consisted of elements relating to part transportation, routing and identification and a manufacturing process module responsible for the product transformation processes.

3.5. Step 4 – Evaluate Module Concepts

The modularity concepts developed in the previous step were then evaluated in order to determine the final modular architecture for both the cyber and physical aspects of the PS. This evaluation was carried out using a decision matrix method. The different concepts were scored on criteria such as ease of implementation, technical feasibility, security and safety. The selected cyber and physical modular system architecture solutions are illustrated in Fig. 6and Fig. 7.

Step 5 – Improve Each Module

Design for modularity approaches mainly based on Ulrich and Tung [37] five different categories of modularity, were then employed to improve the modules and their interfaces. Mainly a bus modularity system was employed. For the modular cyber architecture, it was decided that the “bus” module would be the non-deterministic communication protocol which would allow all other modules to communicate and pass on data between them.

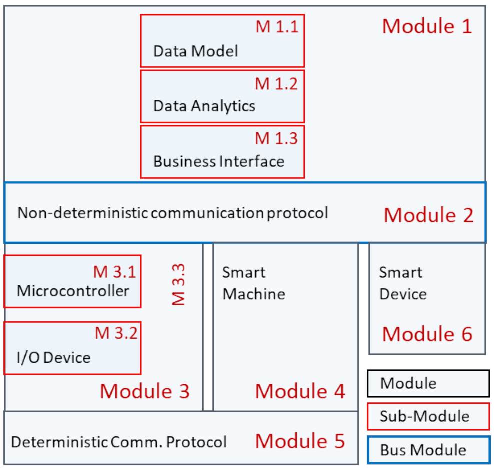

Fig. 6 - Modular cyber system architecture

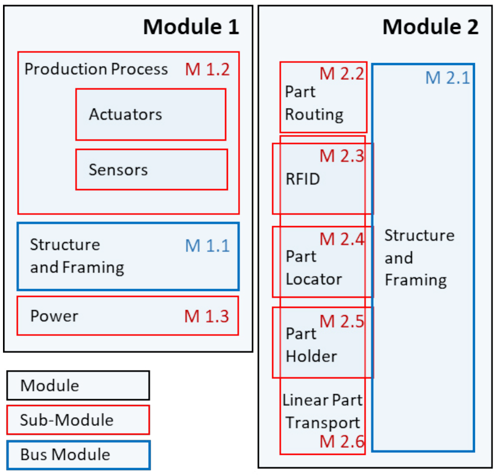

Fig. 7 – Modular physical system Architecture

For the modular physical architecture, the structure and framing of the system was selected as the “bus” module. This would allow all the other modules to be plugged on or attached to this base module. This therefore required the selection and development of the right interfacing solutions between the modules and the main bus module. Details of how this was implemented will be provided in the next section.

4. Modular CPPS Implementation

The aim of implementing a modular CPPS was to help the researchers in this study understand better the elements required to implement a CPPS, and also to understand the specific challenges brought about by modularity. Furthermore this implementation demonstrates and explains in practice how the modular cyber and physical system architectures developed using the MFD method can be implemented.

4.1. Implementation of Cyber Modular Architecture

The implementation of the cyber modular architecture is illustrated in Fig. 8. The main bus is Module 2, which is based on the OPC/UA communication and utilizes an IoT gateway to interface with Module 1. Module 1 is based on cloud computing and was implemented using the Microsoft’s Azure cloud platform and IoT hub. This provided the possibility of implementing sub functionalities of data storage, data analytics and business processes.

Module 3 is a Smart controller which implements the concept of a decentralized control system. This smart controller was designed to be responsible for carrying out independent logical function execution and input/output signal operations. This module was implemented via a Siemens s7-1200 logic control on which an OPC/UA server was set up in order to transmit and receive data from the cloud. Module 4 is a smart machine which can communicate with the cloud via OPC/UA and also using the deterministic network (Module 5), which was implemented using the Profinet protocol.

Other modules such as smart devices (Module 6) can be added to the CPPS and retrieve and analyze data within the CPPS by using the node-red javascript development language.

This modular system architecture allows for new modules, such as smart controllers, machines or devices to be added to the CPPS, as long as they can connect via OPC/UA and data and instructions can be passed on via this main bus. Safety dependent devices are connected via the Profinet protocol to ensure reliable and safe communication. The Azure cloud platform also allows for internal modularity since different cloud applications can be developed, added or removed to the cloud dependent on the CPPS requirements.

4.2. Implementation of the Physical modular architecture

The physical aspect of the CPPS was also constructed based on the modular architecture presented in the previous section. The main challenge for the physical aspect was to develop interfaces between the modules which allowed the possibility for modules to be added and/or removed from the system. Another challenge was the alignment of the part which was being transported via the material handling module (Module 2), with respect to the process module (Module 1). Fig. 9, illustrates how the physical modular architecture was implemented into a CPPS.

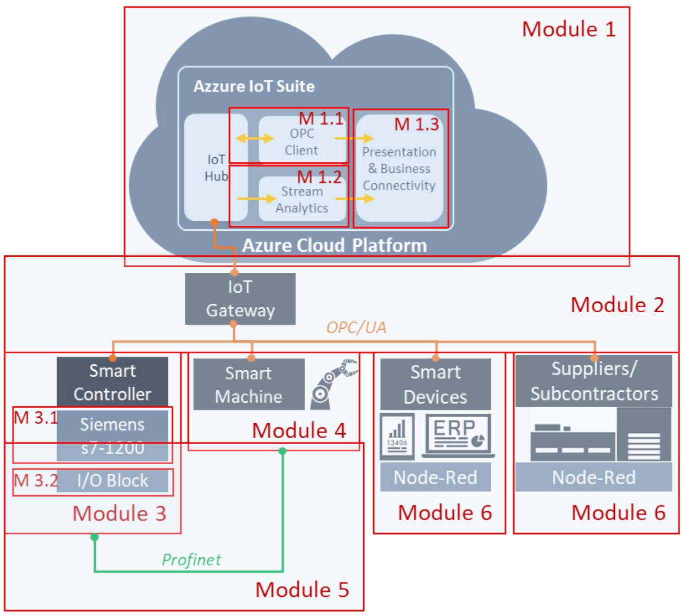

Fig. 8 - Implementation of cyber modular architecture

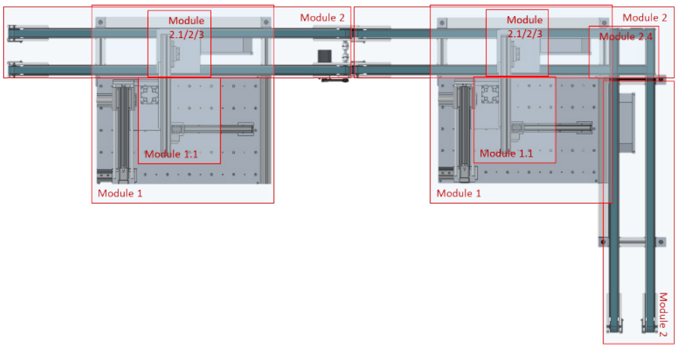

Fig. 9 - Implementation of the physical modular architecture

One feature of note is the drilled base plate which forms part of the structure and framing of Module 1. This allows for production process sub-modules (1.2) that contain the actuators and sensors required for the transformation process to be easily added and removed to the CPPS.

5. Conclusions

Modularity is a fundamental requirement for Cyber-Physical Production System implementation. This research has therefore contributed a modular system design approach for CPPS, based on the MFD method. This approach was utilized to develop a modular system architecture for both the cyber and physical aspects of CPPS. A CPPS was then implemented based on this modular system architecture. Future work will continue developing this method and the CPPS presented here by exploring different modularity concepts and by developing further the interfaces between these modules.

Acknowledgments

The authors would like to thank the University of Malta for the financial support through the Research Grant "Digital Planning and Simulation for the Factory of the Future" (Vote No. IMERP 05-17).

References

[1] Monostori L. Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. Procedia CIRP 2014;17:9–13.
[2] Lee EA, Seshia S. Introduction to Embedded Systems - A Cyber-Physical Systems Approach. Second Edition. UC Berkeley: 2015.
[3] Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M. Industry 4.0. Bus Inf Syst Eng 2014;6:239–42.
[4] Vadde S, Kamarthi SV, Berry NM. Smart sensors in a flow-line manufacturing system: an agent-based simulation. Int. Conf. Integr. Knowl. Intensive Multi-Agent Syst. 2005, 2005, p. 635–40.
[5] Xu X. From cloud computing to cloud manufacturing. Robot Comput-Integr Manuf 2012;28:75–86.
[6] IEC 62541. OPC Unified Architecture, V2.0 2016.
[7] Drath R, Luder A, Peschke J, Hundt L. AutomationML - the glue for seamless automation engineering. 2008 IEEE Int. Conf. Emerg. Technol. Fact. Autom., 2008, p. 616–23.
[8] ElMaraghy HA, Schuh G, ElMaraghy W, Piller F, Schönsleben P, Tseng M, et al. Product variety management. CIRP Ann - Manuf Technol 2013;62:629–52.
[9] Koren Y. Globalization and Manufacturing Paradigms. Glob. Manuf. Revolut., John Wiley & Sons, Inc.; 2010, p. 1–40.
[10] ElMaraghy HA, editor. Changeability Effect on Manufacturing Systems Design. Chang. Reconfigurable Manuf. Syst., London: Springer London; 2009.
[11] Schenk M, Wirth S, Müller E. Factory Planning Manual. Springer; 2009.
[12] ElMaraghy W, ElMaraghy H, Tomiyama T, Monostori L. Complexity in engineering design and manufacturing. CIRP Ann - Manuf Technol 2012;61:793–814.
[13] Westkämper E. Digital Manufacturing in the Global Era. Digit. Enterp. Technol., 2007, p. 3–14.
[14] Francalanza E, Borg J, Constantinescu C. A Knowledge-based Tool for Designing Cyber Physical Production Systems. Comput Ind 2017;84:39–58.
[15] Phillips E. Manufacturing Plant Layout. Society of Manufacturing Engineers,U.S.; 1997.
[16] Chryssolouris G. Manufacturing Systems: Theory and Practice. Springer; 2006.
[17] Rittel HWJ, Webber MM. Dilemmas in a general theory of planning. Policy Sci 1973;4:155–69.
[18] Francalanza E, Borg J, Constantinescu C. Approaches for Handling Wicked Manufacturing System Design Problems, Gulf of Naples, Italy: 2017.
[19] Ueda K, Markus A, Monostori L, Kals HJJ, Arai T. Emergent Synthesis Methodologies for Manufacturing. CIRP Ann 2001;50:535–51.
[20] Rogers GG, Bottaci L. Modular production systems: a new manufacturing paradigm. J Intell Manuf 1997;8:147–56.
[21] Schuh G, Lenders M, Nussbaum C, Kupke D. Design for Changeability. Chang Reconfigurable Manuf Syst 2009:251–266.
[22] Nuzzo P, Sangiovanni-Vincentelli AL, Bresolin D, Geretti L, Villa T. A Platform-Based Design Methodology With Contracts and Related Tools for the Design of Cyber-Physical Systems. Proc IEEE 2015;103:2104–32.
[23] Hankel M. The Reference Architectural Model Industrie 4.0 (RAMI 4.0) 2015.
[24] Uslar M, Engel D. Towards Generic Domain Reference Designation: How to learn from Smart Grid Interoperability. Conf. --Ch Energieinformatik 2015 Karlsr. Karlsr. Ger. Vol. 1, 2015.
[25] Stark R, Kind S, Neumeyer S. Innovations in digital modelling for next generation manufacturing system design. CIRP Ann 2017;66:169–72.
[26] Onori M, Semere DT, Lindberg B. Evolvable Systems: An Approach to Self-X Production. In: Huang GQ, Mak KL, Maropoulos PG, editors. Proc. 6th CIRP-Spons. Int. Conf. Digit. Enterp. Technol., Springer Berlin Heidelberg; 2010, p. 789–802.
[27] Schleipen M, Lüder A, Sauer O, al et. Requirements and concept for Plug-and-Work: Adaptivity in the context of Industry 4.0. Autom AT 2015;63:801–20.
[28] Onori M, Lohse N, Barata J, Hanisch C. The IDEAS project: plug & produce at shop-floor level. Assem Autom 2012;32:124–34.
[29] Yusuke Maeda, Haruka Kikuchi, Hidemitsu Izawa, Hiroki Ogawa, Masao Sugi, Tamio Arai. "Plug & Produce" functions for an easily reconfigurable robotic assembly cell. Assem Autom 2007;27:253–60.
[30] Theorin A, Bengtsson K, Provost J, Lieder M, Johnsson C, Lundholm T, et al. An event-driven manufacturing information system architecture for Industry 4.0. Int J Prod Res 2017;55:1297–311.
[31] Zuehlke D. SmartFactory—Towards a factory-of-things. Annu Rev Control 2010;34:129–38.
[32] Fisher K. Agent-based design of holonic manufacturing systems. Robot Auton Syst 1999;27:3–13.
[33] Wiendahl HP, Greim H, Nyhuis P, Zäh M, Wiendahl H, Duffie N, et al. Changeable Manufacturing - Classification, Design and Operation. CIRP Ann - Manuf Technol 2007;56:783–809.
[34] Simpson TW, Siddique Z, Jiao RJ, editors. Product Platform and Product Family Design: Methods and Applications. Springer; 2005.
[35] Erens F, Verhulst K. Architectures for product families. Comput Ind 1997;33:165–78.
[36] Erixon G. Modular Function Deployment: A Method for Product Modularisation - Gunnar Erixon - Google Books. PhD. Royal Inst. of Technology, Department of Manufacturing Systems, Assembly Systems Division, 1998.
[37] Ulrich KT, Tung K, Management SS of. Fundamentals of product modularity. Sloan School of Management, Massachusetts Institute of Technology; 1991.