Multi-agent Interaction Structure for Enabling Subsidiary Planning and Control in Modular Production Systems

Simon Komesker[1,2](✉), Jonathan Bartels[1,2], Achim Wagner[3], and Martin Ruskowski[1,4]

[1] TU Kaiserslautern, 67663 Kaiserslautern, Germany
simonkomesker@gmx.de
[2] Volkswagen AG, 30884 Wolfsburg, Germany
[3] Deutsches Forschungszentrum Für Künstliche Intelligenz GmbH, 67663 Kaiserslautern, Germany
[4] Technologie-Initiative SmartFactory KL E.V, 67663 Kaiserslautern, Germany

Abstract. Modular production systems enable resilient production processes through decoupled production processes. On the way to implementing flexible and adaptable production systems, information support plays a decisive role. Only the use of intelligent and structured information processing across previous system boundaries and areas enables the coordination of requirements and capacities in dynamic production environments. The rigid communication structures in information systems of current production systems therefore need to be replaced by dynamic interaction, both horizontally between entities and vertically between different hierarchical levels. Multi-agent systems (MAS) are one way to meet the requirements for centralized and decentralized decision making in complex (cyber physical production) systems (CPPS). To prepare the instantiation of a MAS, it is necessary to structure and describe the information flows of a production system.

In this paper, the results of a simulation experiment for the implementation of collaborative, subsidiary decision making based on a model-based system structure are presented. Productivity potentials of more than 10% can be shown by using collaborative manufacturing strategies.

Keywords: modular production · multi-agent systems · interaction model · systems architecture

1 Introduction

Resilient production systems guarantee robust production processes in the event of unforeseen deviations in the operating sequence. Kern's Modular Production proves to be more resilient to demand and capacity fluctuations compared to linear production systems [1, 2]. With the use of Cyber Physical Systems (CPS), mechanical and mechatronic elements in (socio-technical) production systems acquire a higher decentralized decision-making capability and possess complex interaction [3]. Systems engineering

N. Kiefl et al. (Eds.): SCAP 2022, ARENA2036, pp. 354–364, 2023.
https://doi.org/10.1007/978-3-031-27933-1_33

principles enable a better understanding and designing of complex phenomena [4]. Thus, a system structure is needed that enables structured information processing in modular production systems [2].

The requirements for information processing in modular production systems with alternative process design according to Kern have been identified in former research [2, 5]. This paper follows up on these and validates the previous research in the form of a simulation experiment. For this purpose, the related work of preparing structured information processing in multi-agent systems (MAS) for matrix production like Kern's Modular Production is given in Sect. 2, as well as the basics of systems engineering. In Sect. 3, the basic functions for operating a MAS in modular production systems are identified and put together in an interaction model. This model is supported with a system structure in Sect. 4. Section 5 evaluates the implementation before the paper is concluded in Sect. 6.

2 State of the Art and Related Work

In the following section, the related work in the area of systems structures for cross-system information processing is presented.

2.1 Structured Information Processing in Modular Production

Modular Production Systems require cross-system information processing [4]. The control paradigm of strictly hierarchical control, e.g. the ISA95 is not suited for dynamic and flexible cross-system information processing [6]. Alternative control paradigms that allow vertical and horizontal communication are hybrid control paradigms [7]. A promising approach to enable hierarchically but flexible information processing is the application of multi-agent systems, that allow for complex communication between encapsulated agents as subsidiary [8, 9]. As previous research stated, a special focus lays upon the systems architecture and structure to support the requirements for information processing between different domains [2, 5]. There are existing architectures that serve as orientation for MAS development [10, 11]. An approach preferably suited for reconfigurable MAS are AOSE methods like ADMARMS which is based on high level design principles and a rather functional oriented development [10].

The coordination of systems is based on interaction of product, processes and resources in CP(P)S [12]. Different possibilities exist to apply coordination by planning and scheduling [13]. These could be used to coordinate systems and its inner systems and elements on different hierarchy levels.

2.2 Structuring Systems for Developing a Systems Design

A System can be seen as a combination of interacting elements organized to achieve one or more stated purposes [4]. Systems Engineering is focused on the system as a whole, it looks at the system from the outside as well as from the inside using different principles and concepts [14]. The functional concept describes the functions of a system and what a system does. The structural concept describes the interior of the system, as well as the relationships of elements within this system. Elements of a system can also represent systems on a lower level (system of systems) [15]. This nested relation can be described by the hierarchical concept [4, 16]. A subdomain is the systems design, that incorporates the architectural, logical and physical setup. By using a model-based architecting approach a suitable support of informational processes can be created [17, 18].

In addition technical cybernetics analyze the information flow through the system and how it is processed. It considers how this can be used to manage and control itself as a control loop [19].

3 Modelling Interaction for Planning and Control in Modular Production Systems

An Industrie 4.0 compliant systems structure for modular production systems needs to support complex decision making [5].

The procedure for decomposing a system top-down follows a basic principle "From the General to the Detail" [16]. A decomposition is carried out to managerial and operational independence of functions to create a functional architecture [18]. The coordination of functions follows the bottom-up principle and couples different functional blocks to process (steps). As part of the systems analysis, the system's elements are identified as well as the tasks, roles and interdependencies for a system design view [5]. The application of ADMARMS design methodology supports a MAS architecture with strong focus on maintaining the independence of functional requirements [10].

3.1 Top-Down Decomposition and Bottom-Up Aggregation of a Modular Production System

The system is basically designed as a fractal structure with similar subsystems on different hierarchy layers. The subsystems are designed as encapsulated entities, which can be supported by the concept of AAS and ASD [5] (see Fig. 1).

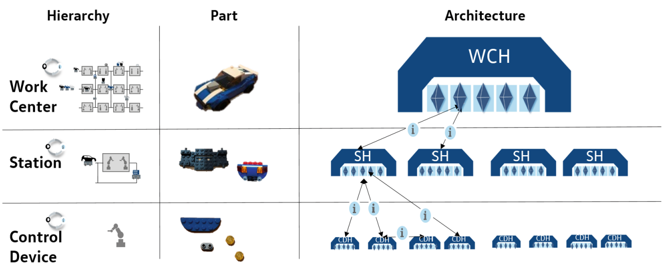

Fig. 1. Decomposition of the production system for structured information processing for a holonic architecture

The structuring was done by defining hierarchy layers for the Modular Production by Kern according to the RAMI4.0 levels followed by a functional requirements engineering matching the manufacturing system design. Each system contains similar functional blocks that are encapsulated as holons on different hierarchy levels according to the RAMI4.0 and preparing a subsidiary decision making.

The challenge of modular production systems was identified in the information processing for a resilient production flow in skill-based modular production systems. The functions and basic elements of a system from an information point of view were extracted based on requirements on previous research [2, 5]. They are separated as representation and self-description functions on the one hand and coordination functions on the other:

I Representation and Self-description Functions

The **Resource Agent (RA)** knows capabilities and availability status of an element or system. It includes skillset, process times and setup matrix. The capabilities or skills are aggregated bottom up from each level to fulfil orders and suborders. The **Product Agent (ProdA)** gives an overview of all manufacturing steps for a specific product in a subsystem. It defines configuration, quality requirements and start and end dates for each item in the production program. It contains information about the order and the manufactured product, with update information from the resources for completed process steps. This interacts with the shell that contains order specific blank options for the product being built. The material is available in sufficient quantity in the supermarket and the transportation is coordinated with the production flow. The production flow manager need the information of material availability for the planned production program. The **Quality Agent (QA)** evaluates quality data from products and parts/material. The **Process Agent (ProcA)** describes skills of the production processes needed to manufacture products with suitable resources.

II Coordination Tasks

The **Production Flow Manager (PFM)** ensures, that the right processes are performed on the right resources in the right time. Therefore the PFM needs information about the product process steps, the precedence graph and the offers from the resources for processing orders and suborders. The PFM schedules and reschedules the orders on the different resources. The **Production System Manager (PSM)** ensures that the right resources are available to produce the production program. A potential measure could be the reconfiguration of a system e.g. a resource adaption or integration of unplanned orders, because the (sub-)system matches the required skills for that order. KPIs to consider in this context are transportation time, variant flexibility, value add time, setup time, makespan, output. The **Material Agent (MA)** ensures, that the right material is at the right place at the right time and the right volume The **Data Manager (DA)** collects all the actual and requested information in the system and provides a consistent data base. The **Deviation Agent (DevA)** identifies and assesses deviations initiates activities in the system. The mechanism for planning and scheduling is presented in [5] and defines a subsidiary decision making process for multi-level modular production systems. The **Orchestration Agent (OA)** is responsible for the execution of decisions and closing the control loop of an integrated planning and control system.

3.2 Interaction Model for Cross-System and Cross-Level Information Processing

Based on the identified agent functionalities, an interaction model was developed as basis for a functional architecture in the sense of a model-based architecture [15]. The formalized interaction model derived from the functional description in Sect. 3.1 is depicted in Fig. 2. The interaction model is used for every fractal system of the production system to support its subsidiary decision making. A fractal is defined per hierarchy level as a work center, a station or a resource, each represented as a holon (see Fig. 1). The interaction is described on an abstract level as control/command and inform. Control/command interactions need a confirmation of receive, an inform a typical message into an agents inbox. Asynchronous horizontal and vertical communication is realized by the DA that serves as a message broker between agents, structuring data in different topics to which other agents are subscribed and therefore enabling cascaded feedback control loops.

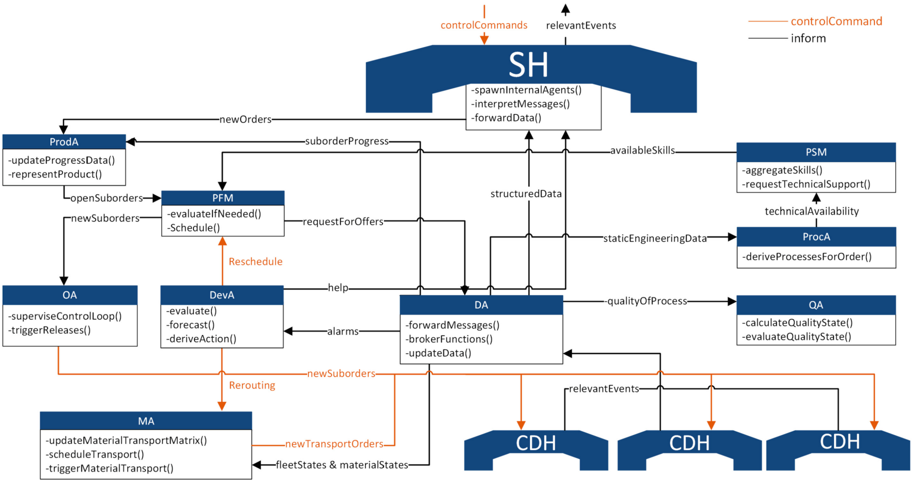

Fig. 2. Interaction model for multi-level and cross-system holistic planning and control

4 Implementation of a Subsidiary Planning and Control for Enabling a Resilient Production Flow in Modular Production

In this section a system structure for a use case is implemented based on the interaction model for subsidiary decision making. This is done by using the functional decomposed blocks and aggregate it bottom-up in a multi-level production system.

4.1 Introduction of the Use Case

The production system contains 84 resources to produce 3 different models of a car with different but similar product structure. The system is structured hierarchically in 3 levels according to RAMI 4.0 consisting of fractals designed as holons with the same agent architecture. The **work center level**, that schedules and orchestrates the production of the product. The **station level**, that schedules and orchestrated the production of modules of the product in stations. And **the control device level**, where the processes to manufacture parts for modules are being executed the scheduled sequence. The control device level marks the lowest level of granularity of functions for operational and managerial independence. The production program consists of 300 products, each of which consists of 16 modules and the associated production orders, 3 of which can be manufactured in parallel. The module orders contain 2 to 8 production orders, so that the total number of process steps for a product amounts to up to 90 steps. A data and information model is supporting the function of a skill-based coordination and allocation of the different product needs and resource capabilities with the respective process skills.

4.2 Implementation of a Model-Based System Structure

To create the system structure, the interaction model is prototyped as a holonic MAS. The intelligent medium- and long-term decision-making is realized with the help of agent-based modeling and simulation (ABMS). The coupling with a production system is realized using the industrial-grade Discrete Event Simulation (DES) tool Plant Simulation, which is coupled via a middleware with the ABMS and a scheduling module.

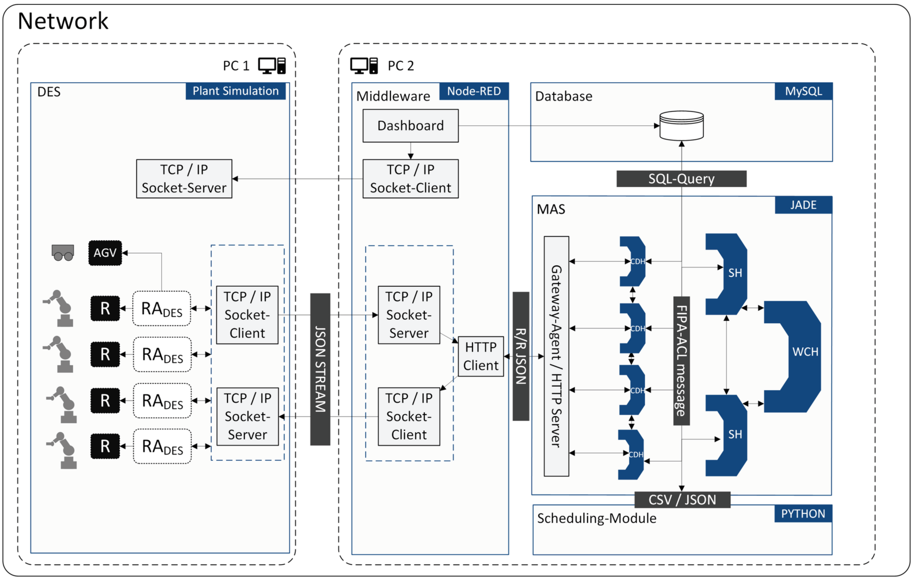

Fig. 3. Implemented system architecture from of the interaction model and planning mechanism

The **DES** provides the data basis for the decisions as a simulated production system. Based on the skill-based approach, the resources have the decomposed skills to execute process steps to manufacture a part, module or product. The formerly passive entities of the DES were agentified in order to be able to communicate horizontally and decide decentrally for ad-hoc initiation of transport orders and execution of short-term alternative strategies. The material transport is modeled by freely moving AGV within the station and by lane-bound AGV between the stations.

By using a low-code **Middleware** and TCP/IP based interfaces a synchronous communication between the DES and the MAS can be established. The goal is to connect each passive resource of the DES with an active equivalent in the MAS. The middleware forwards the event-based JSON-formatted updates from the DES to an http-enabled gateway agent of the MAS via the TCP/IP client/server socket interface and returns derived actions, as shown in Fig. 3. Via a web-based user interface, the middleware enables configuration of the MAS, scheduling module and DES.

The **MAS** is the digital representation of the production system and structures the agents within it based on the interaction from Fig. 2. The MAS is a multi-agent system. In implementing the interaction model, the focus was on cross-level processing and functions were aggregated. Based on the FIPA-ACL compliant agent platform JADE, a MAS was developed that implements the sequencing and allocation planning [4]. For this purpose, each control-device level resource was implemented as a holonic CDH in a 1:1 relationship in the MAS. In the prototypical case, each Holon encapsulates several of the agents presented in Sect. 3, within the respective system as a black box. These include the CDH representation function and the DM's function, which interacts with an SQL database. In addition, the function of checking the available skills of resources of the PSM is carried out during initialization and failure events. In the case of level-specific sequence and allocation scheduling, a scheduling module is triggered in each case, corresponding to the function of the PFM. Through the interaction of individual encapsulated holons as well as the cross-level decomposition of orders, the holons solve the problem of sequence and allocation scheduling subsidiary and forward the resulting plans as executable orders for implementation to the executing control device level of the DES.

The **Scheduling Module** serves as an implementation of the PFM for the subsidiary coordination of resources, processes and products. The scheduling problem of the use case describes an extended flexible job shop problem. The scheduling module for solving the problem was implemented in Python and solved using a heuristic based on the Tabu-Search method. The holons use the module to perform an allocation of the hierarchically distributed operations of a task with the resources available in the respective holon. The agents have the possibility to use the heuristics of the module when requesting an offer as well as to optimize the final schedules by the heuristics and to forward concrete orders based on them. The scheduling module optimizes the schedule according to the lead time.

The developed system structure represents a production system as **a system of systems,** which is able to solve subsidiary decisions on a short-term decentralized shop-floor level as well as medium- and long-term planning problems through the coupled MAS. After an initial decomposition of the planning problem in the MAS, a sequence plan is generated by the holons communicating with each other. The final production orders are derived and forwarded to the DES resources using the middleware. The ad-hoc transport planning as well as the short-term decentralized reaction of the resources to disruptions is realized by the horizontal communication in the DES.

5 Simulation Setup and Results

In this section, the configuration and the corresponding execution of the simulation runs are explained in more detail. In a first step, the different variables are presented as well as the evasion strategies of the resources on control-level-device and station-level within the simulation runs.

The overall layout of the DES, consisting of 84 control-devices containing skills grouped into 16 stations in 1 work-center is designed as a matrix. The production program of 300 products and three types is initially planned for 110 products and then released in groups of 30 products when the number of products in the system is lower than 80 products. The buffer capacity of resources on control-device-level is limited to 2 with an availability of 95% with an MTTR of 2:30. The transport within each station is realized by 5 AGV, for inter-station transport with 40 AGV. The station buffer capacity is set to 20.

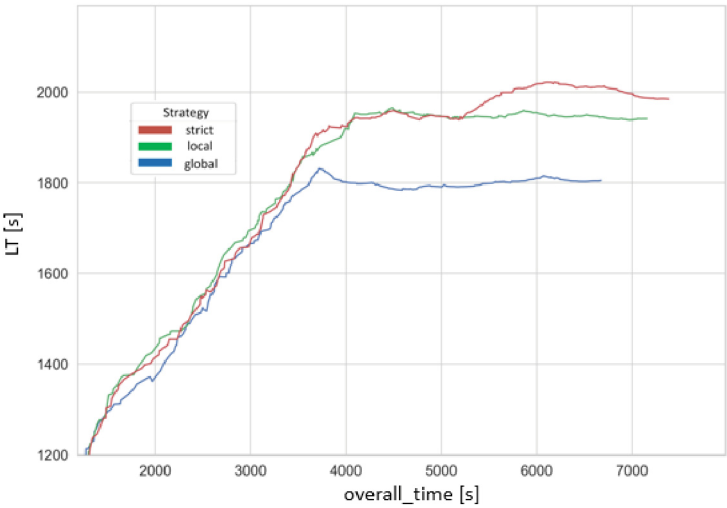

Fig. 4. Exemplary graph of three different simulation runs following different manufacturing strategies

The simulation runs follow different manufacturing strategies to show the effects of different collaborative decentral and central control mechanisms. Three strategies *strict (strict plan execution), local (local negotiation for alternatives) and global (global negotiation for alternatives)* were followed to simulate different situations of deviation events at control device or station level. The *strict* strategy follows the initial plan from the scheduling tool and does not allow alternative process sequences. The *local* strategy allows to reassign a product in case a resource failure occurs. In this case, the alternatives that the technical precedence graph allows are evaluated. Either the process step can be completed in the same station at a different resource with the required skill or a different order with a different skill requirement can be assigned. Compared to the *local* strategy, the *global* strategy additionally allows the selection of another station within the overall system to process the open order by using the agent interaction for negotiation of alternative resource allocation and product sequences.

While the lead time (LT) for 300 products of the *strict* simulation runs is 7670 s on average, the LT decreases to 7313 s in case of the *local* strategy. The *global* strategy reduces to an averaged 6891 s. The transportation time percentage (AVG_TP) with the *global* strategy of all products are on average 2.8% higher than with the *strict* strategy. At the same time, the storage time percentage (AVG_SP) of the products are reduced by 5.8%, which results in an overall increase of the percentage of value-added production time (PP) by 3%. Comparing the lead times for the mentioned production program, this results in an average reduction of the LT of 11.2% for the same product mix and the same incident volume. Figure 4 shows an exemplary graph of lead time of products LT leaving the system in an exemplary graph. In the start-up phase of the system, the strategies do not differ due to low competition for resources. After the ramp up phase, the saving corresponds to about 11.2% (Table 1).

Table 1. Results of average makespan [overall_time], transportation time percentage [AVG_TP], average storage time percentage [AVG_SP], average value-added production time percentage [AVG_PP] and average leadtime [AVG_LT] of the different manufacturing strategies in 27 simulation runs

Strategy	overall_time [s]	AVG_TP [%]	AVG_SP [%]	AVG_PP [%]	AVG_LT [s]
global	6891.108	13.9	57.3	28.9	1890.744
local	7313.016	12.6	59.9	27.6	1986.008
strict	7670.065	11.1	63.2	25.9	2122.766

6 Conclusion

This paper shows a system structure for holistic information processing with centralized and decentralized intelligence. This is based on an interaction structure for cross-system (horizontal) and cross-level (hierarchically) interaction for an autonomously organized production flow. A hybrid control paradigm was implemented using MAS and shows its potential for the use of centralized and decentralized planning and control in a use case of a modular production systems. This allowed the implementation of an agent-based planning mechanism that realizes sequence planning over multiple levels. The planning and control mechanism was designed using an interaction model for a holistic support. For the execution of the planning and control in dynamic environments a hybrid decision support is needed, which was prototypically implemented in a simulation experiment. The interaction model was implemented focusing on cross-level interaction and collaborative decision making between holons of different systems in a MAS. The collaborative decision making based on cross-level and cross-system interaction in a holonic MAS enables an optimization of the lead time by more than 10% for dynamic production situations. Agent based scheduling and control is used in different scenarios in the experiment. Firstly, a global optimal schedule for a multi-level production flow was created by negotiating between different local optima in the manufacturing system. Secondly, during manufacturing execution the agents optimize the global production flow autonomously by cross-system and cross-level interaction in system. By using the interaction structure, an alternative production flow is created and orchestrated, which solves unforeseen events e.g. deviations caused by machine failures. In future experiments, the interaction structure will be thoroughly validated by adding additional agents to the holonic architecture. For further research, additional scheduling methods can be applied as well as further enhancements of the cascaded control loops to allow a more dynamic and predictive assessment of deviations and possible risks.

Acknowledgements. This work was supported by the European Union under the "Horizon 2020" research program (Grant no. 957204) within the project "Multi Agent Systems for Artificial Intelligence" (MAS4AI).

References

1. Kern, W.: Modulare Produktion – Methodik zur Gestaltung eines modularen Montagesystems für die variantenreiche Serienproduktion. Springer Vieweg. Stuttgart, Univ., Diss, Wiesbaden (2021)
2. Komesker S. et al. Structured information processing as enabler of versatile, flexible manufacturing concepts. In: Weißgraeber P., Heieck F., Ackermann C. (eds.) Advances in Automotive Production Technology – Theory and Application. ARENA2036, pp. 108–116. Springer Vieweg, Berlin (2021). https://doi.org/10.1007/978-3-662-62962-8_13
3. Kagermann H, Wahlster W, Helbig E.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Abschlussbericht des Arbeitskreises Industrie 4.0, Bundesministerium für Bildung und Forschung. Frankfurt am Main, Germany: Forschungsunion Wirtschaft und Wissenschaft (2013)
4. Bajzek, M., Fritz, J., Hick, H.: Systems engineering principles. In: Hick, H., Küpper, K., Sorger, H. (eds.) Systems Engineering for Automotive Powertrain Development. P, pp. 149–194. Springer, Cham (2021). https://doi.org/10.1007/978-3-319-99629-5_7
5. Komesker, S., et al.: Enabling a multi-agent system for resilient production flow in modular production systems. Procedia CIRP **107**, 991–998 (2021)
6. Trunzer E. et al.: System architectures for Industrie 4.0 applications: derivation of a generic architecture proposal. Prod. Eng. **13**, 247–257 (2019)
7. Cruz Salazar, L.A., Ryashentseva, D., Lüder, A., Vogel-Heuser, B.: Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns. Int. J. Adv. Manuf. Technol. **105**(9), 4005–4034 (2019). https://doi.org/10.1007/s00170-019-03800-4
8. Ruskowski M. et al. Production Bots für Production Level 4: Skill-basierte Systeme für die Produktion der Zukunft. atp magazin **62**(9), 62–71 (2020)
9. Smart Factory e.V., Production Level 4 – Der Weg zur zukunftssicheren und verlässlichen Produktion, Whitepaper, SF_Whitepaper-Production-Level-4_WEB.pdf (smartfactory.de), Accessed 08 June 2022
10. Farid, M., Ribeiro, L.: An axiomatic design of a multiagent reconfigurable mechatronic system architecture. IEEE Trans. Industr. Inf. **11**(5), 1142–1155 (2015)

11. Valckenaers, P., et al.: Designing Holonic manufacturing systems. Rob. Comput.-Integr. Manuf. **14**, 455–464 (1998)
12. Lueder, A.: Flexibility in production systems by exploiting cyberphysical systems. Computer **53**(1), 81–85 (2020)
13. Leusin, M., et al.: Solving the job-shop scheduling problem in the industry 4.0 era. Technologies **6**(4), 107 (2018)
14. Kossiakoff, A., et al.: Systems Engineering Principles and Practice, 2nd edn. John Wiley, New York (2011)
15. Haberfellner, R., et al.: Systems Engineering Fundamentals and Applications. Springer, Switzerland (2019)
16. Ropohl, G.: Allgemeine Technologie: eine Systemtheorie der Technik. KIT Scientific Publishing, Karlsruhe (2009)
17. Weilkiens, T., et al.: Model-Based System Architecture, 2nd edn. John Wiley, New York (2022)
18. Buede, D.: The Engineering Design of Systems: Models and Methods, 2nd edn. John Wiley, New York (2008)
19. SEBoK Guide to the systems engineering body of knowledge (SEBoK). http://www.sebokwiki.org/w/index.php?title¼Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)&oldid¼453122. Accessed 08 June 2022