CONFERENCE ON PRODUCTION SYSTEMS AND LOGISTICS
CPSL 2023

5th Conference on Production Systems and Logistics

Towards a Service-Oriented Architecture for Production Planning and Control: A Comprehensive Review and Novel Approach

Sebastian Behrendt[1], Florian Stamer[1], Marvin Carl May[1] , Gisela Lanza[1]
[1]wbk Institute of Produciton Science, Karlsruhe Institute of Technology

Abstract

The trends of shorter product lifecycles, customized products, and volatile market environments require manufacturers to reconfigure their production increasingly frequent to maintain competitiveness and customer satisfaction. More frequent reconfigurations, however, are linked to increased efforts in production planning and control (PPC). This poses a challenge for manufacturers, especially in regard of demographic change and shortage of qualified labour, since many tasks in PPC are performed manually by domain experts. Following the paradigm of software-defined manufacturing, this paper targets to enable a higher degree of automation and interoperability in PPC by applying the concepts of service-oriented architecture. As a result, production planners are empowered to orchestrate tasks in PPC without consideration of underlying implementation details. At first, it is investigated how tasks in PPC can be represented as services with the aim of encapsulation and reusability. Secondly, a software architecture based on asset administration shells is presented that allows connection to production data sources and enables integration and usage of such PPC services. In this sense, an approach for mapping asset administrations shells to OpenAPI Specifications is proposed for interoperable and semantic integration of existing services and legacy systems. Lastly, challenges and potential solutions for data integration are discussed considering the present heterogeneity of data sources in manufacturing.

Keywords

Service-oriented production; asset administration shell; service-oriented architecture; interoperability; Industry4.0

1. Introduction

The integration of modern information technologies into production systems, as proposed by concepts such as Industry4.0 [1], promises to be an answer to handle the increasing volatility in markets [2,3] and the need for more efficient, automated and adaptive manufacturing [4]. Although digitalization enables the continuous exchange of information between IT-systems and the shop floor for monitoring or control purposes, many processes in production planning and control (PPC) are still performed either manually or with software that only supports proprietary interfaces and often lacks automation and integration capabilities [5]. The trend towards more volatile, uncertain, complex, and ambiguous markets, as summarized by the VUCA world [6], emphasizes reliable planning as a crucial factor to get into production quickly, especially for low-volume products and complex product portfolios. Concepts such as reconfigurable manufacturing systems enable frequent and responsive adaption of system structure and logic to efficiently meet current customer demand [7,8] . However, frequent reconfigurations also increase the effort in PPC, motivating for digitalization and automation in this domain [9,10].

DOI: https://doi.org/10.15488/15271
ISSN: 2701-6277

In this sense, technological advances show potential for developing and integrating software solutions for planning and control of manufacturing systems that can interact in an interoperable way with each other and with the manufacturing system itself [11,12]. The availability of large amounts of data from a digitized shop floor enable the use of algorithms for analysis and planning, such as realistic simulation models for prognosis and analysis [13], optimization tools for capacity planning or layout planning [14], and machine learning for production control [15,16]. Although existing solutions could serve as components for a more automated and data-driven PPC, integrating them into company software architectures for automation poses a challenge in terms of data integration and algorithmic orchestration due to the variety of data sources [17] and the complexity of PPC [18,19]. To date, such algorithms and models are mostly implemented for specific use cases with proprietary tools and lack integration and generalization capabilities [20,21,5]. For example, material flow simulation models of manufacturing systems are mostly modelled by experts during production planning with manual parameter input and are not maintained and used over the life cycle of the later realized manufacturing system [13].

There have been great improvements in interoperable data exchange through technologies such as OPC UA, MQTT, or the asset administration shell (AAS). However, in sight of the volume and variety of data sources in manufacturing [17], a software architecture is needed that is manageable at large scales and prevents data silos and incompatibilities. The software-defined manufacturing (SDM) paradigm suggests to decouple software applications from the underlying manufacturing infrastructure through a control layer that simplifies the orchestration of information flows by abstraction, virtualization, and interface unification [22,23]. Developing solutions for tasks in PPC based on unified interfaces either through standardization or integration promises to increase their reusability and interoperability.

A related principle that is widely promoted in the areas of Smart Manufacturing [24], Cyber-physical Production Systems (CPPS) [11], Industry4.0 [1] or Cloud Manufacturing [25] is service-orientation. It is already the leading principle in many technologies such as OPC UA or web services and has emerged as a suitable architecture for a highly scalable orchestration of software systems [26]. In a software-oriented architecture (SOA), software components, i.e. services, are self-contained and modular units of functionality that communicate via standardized interfaces [27]. Due to the separation of interface and implementation, these services can interact with each other in complex scenarios without the need for mutual understanding of their underlying functionality [27]. The capabilities of SOA in terms of adaption, abstraction, and integration make it a suitable architecture to realize SDM in the domain of PPC. Therefore, we investigate how the principles of service-orientation can be applied to tasks in PPC to improve their potential for reusability, automation, and integration in an appropriate software architecture.

In the following, Section 2 reviews existing approaches in the literature that aim at digitalization of PPC and assesses their degree of service-orientation and integration capabilities. In Section 3, we will present an approach that aims at the servitization of PPC by describing PPC tasks as services with consideration of data integration theory. Moreover, we present a software architecture that considers SOA, SDM principles, and state-of-the-art technologies for a flexible, automated and data-based PPC. Thereby, special attention is given to the integration of existing legacy systems in manufacturing. Finally, Section 4 concludes the paper and provides an outlook for further research.

2. Related work

The goal of SOA is to organize networks of software systems in large scales while maintaining a flexible and maintainable system that enables interoperable data exchange between components [26]. According to Valipour et al. [27], services comprise of 3 components, i.e. protocol, interface, and implementation, and must implement the following 6 characteristics in SOA:

- **Discoverable and dynamically bound**: services can be discovered by consumers in registries

- **Self-contained and modular**: services are modular components that encapsulate specific functionality
- **Interoperable**: services are able to communicate and interact with each other using standard protocols and data formats
- **Loosely coupled**: a low and well-known number of dependencies between service consumers and service providers
- **Location transparent**: location of services is provided by registries at runtime allowing for simple load balancing
- **Composable**: services can be combined to applications or orchestrations due to their modular structure.

In order to evaluate existing concepts and approaches for digital planning and control of manufacturing systems, we classify their degree of service orientation based on the fulfilment of the aforementioned design principles of SOA. The literature review was thereby conducted based on the methodology of Mayring [28].

Services provide a distinct description how to interact with them by specifying their interface and protocol. The protocol denotes thereby how to interact with the service, e.g. covering authentication or payment, and the interface defines all possible requests of the service and their associated schema, i.e. the input and output data format of a request. However, data still needs to be right in the right format, i.e. data conforms the schema of these interfaces, to ensure interoperability when interacting with a service. Especially in manufacturing, interoperability is challenging due to legacy systems and a large variety of data sources [29,30]. The field of data integration deals with this problem by combining data from different sources to a unified view for the user [31]. Formally speaking, data integration can be described by the tuple $\langle G, S, M \rangle$, where a global schema G, i.e. the unified view, is obtained by applying a mapping M to a source schema S. The mapping M, responsible for translating between schemas S and G, aims to handle with data heterogeneity, both in terms of notational and conceptual heterogeneity [32]. Notational heterogeneity concerns communication protocol or language whilst conceptual heterogeneity is summarized by differences in schema or semantics of the used data models.

Besides the degree of service orientation, we assess the integration capabilities of existing approaches in the field of digital manufacturing based on the degree of data heterogeneity they can manage. In this literature analysis, we will not cover abstract architectures, like the RAMI4.0, as they miss the technical depth for a direct realization, as stated by Wang et al. [33]. Instead, we will concentrate on approaches that are either implemented and evaluated in case studies or concepts that provide enough detail for assessing their fit for a service-oriented PPC.

Lee et al. [34] motivate for the use of CPPS in Industry4.0 to control production operations and present therefore a 5-tier architecture, consisting of the levels: connection, conversion, cyber, cognition, and configure. They emphasize the importance of a tether-free method for exchanging data from various sources with consideration of needed data transformations. The architecture covers with its levels all relevant features of a CPPS but misses detail how to transfer this architecture directly to SOA and implement it with current technologies. A similar approach is proposed by Pérez et al. [35] with a model-based architecture for CPPS that enables vertical integration from the shop floor to the cloud. In this approach, they utilize existing standards and information models to represent manufacturing entities and exchange this information via OPC UA-based services.

Liu et al. [36] present and demonstrate a framework for CPPS that aims for reconfigurability of digital production systems by the use of digital twins and remote control. The authors motivate for a SOA to implement their CPPS and realize it with webservices and MQTT. To limit their efforts in regard of data integration, they use one unified domain ontology that ensures conceptual data homogeneity. A conceptually similar but technologically different approach is shown by Ye et al. [37] based on AAS for interoperable data exchange between cloud and edge software components. They utilize OPC UA for communication to assets and organize applications as AAS-based webservices. The authors note the importance of semantic integration of data for interoperability and ease of integration.

Table 1: Overview of approaches for architectures of digital manufacturing with consideration of production planning and control and service-orientation

	Service-orientation						**Data heterogeneity**	
Approach	Discoverable	Modular	Interoperable	Loosely Coupled	Location transparent	Composable	Notational	Conceptual
Lee et al. (2015) [34]	○	◐	◐	◐	○	○	◐	○
Pérez et al. (2015) [35]	○	●	◐	○	○	○	◐	○
Liu et al. (2020) [36]	○	●	●	○	○	◐	◐	○
Ye et al. (2021) [37]	◐	●	◐	●	●	●	◐	○
Wang et al. (2020) [38]	◐	●	◐	●	●	●	◐	◐
Biesinger et al. (2019) [39]	○	○	●	○	○	◐	◐	◐
Qiu et al. (2007) [40]	○	●	◐	●	○	○	◐	◐
Grassi et al. (2020) [41]	○	●	◐	◐	○	◐	◐	○

To enable digital PPC, Wang et al. [38] motivate to integrate different enterprise application, such as enterprise resource planning (ERP) or manufacturing execution systems (MES) for more collaborative and synergetic information exchange. In their proposed architecture, they utilize industrial internet of things (IIoT) for an interoperable information exchange with production resources. They emphasize the potential to disassemble monolithic applications into services to increase reusability and ease of integration. Yet, a clear explanation how data integration is performed, is missing.

Biesinger et al. [39] concentrate on this data integration by utilizing an enterprise service bus as a central integration entity that allows to connect to heterogenous data sources on the shop floor. They show how to utilize dedicated parsers to extract information in real-time from these data sources and integrate this information in other software systems.

Another approach is introduced by Qiu et al. [40] with a 3 layered architecture where production resources and their controllers are integrated by a service-oriented integration framework with enterprise business applications for planning and control of production. They use pre-defined data formats and protocols to perform the factory integration of heterogenous data sources by mapping the data to knowledge graphs. The demonstrate the approach in semiconductor manufacturing for process control and recipe management

Grassi et al. [41] are concerned with enabling orchestration of control in digital manufacturing. They argue for a decentralized control approach, that is able to handle complexity of manufacturing systems by abstraction of controllers. Yet, their architecture is a monolithic ERP application, that prevents realization of SOA on the application level.

To summarize the reviewed approaches, they all emphasize the importance of an interoperable and flexible information architecture to realize the potentials of digital manufacturing. Although the approaches follow different architecture paradigms and utilize different technologies, all architectures separate business applications from communication with production resource through a dedicated layer for integration, similar to SDM. Moreover, the approaches motivate for decentralized approaches to realize CPPS.

In recent surveys in the areas of smart manufacturing, cloud manufacturing or CPPS, enabling interoperability and information integration are seen as major challenges in realizing these concepts [42,24,43,44]. Although most of the reviewed approaches agree, they do not consider advanced data integration technologies that are scalable for notational and conceptual data heterogeneity. Instead, most approaches rely on tailored integrations or global data models that enable integration.

Another limitation of existing approaches is their degree of service-orientation. Although most of the reviewed approaches promote SOA, they fail to implement essential characteristics. Service discoverability and composability, that is important for SOAs to scale, are mostly not considered. Additionally, compatibility and consideration of existing SOA technologies is mostly not achieved. Lastly, the interoperability of approaches is mostly also limited since only one communication technology, such as OPC UA, is considered. To resolve existing limitations, the remainder of this paper aims to present an approach that considers all described key characteristics of SOA and allows to handle data heterogeneity considering notation but also concepts.

3. Approach to realize service-orientation in production planning and control

The approach considers at first (Section 3.1) how tasks associated with PPC can be described in terms of SOA. Building up on this logic, we describe a software architecture (Section 3.2) that implements the principles of SOA and allows for all data exchanges required to automate PPC. Section 3.3 gives thereby special notice to data integration in this architecture for integration of legacy systems.

3.1 Service abstraction for production planning and control

The aim of PPC is to efficiently and effectively operate production to satisfy customer demands [45]. To do this, a set of tasks is considered by PPC frameworks that are sequentially performed in iterations whereby each task contributes some aspects to the overall PPC solution. Typical tasks are, for example, capacity planning, shop floor scheduling or material requirements planning (MRP). There exist algorithms or models for many PPC tasks which shows the potential for automation. Yet, implementations are mostly done custom-tailored to specific use cases without possibilities for reuse.

To resolve this problem and also ensure that integration in software architectures is not prevented by incompatibilities, software implementations for PPC tasks should follow the design principle of services. Valipour et al. state that the "[…] most important aspect of SOA is that is separates the service's implementation from its interface." [27] This infers for PPC that the data required and produced when solving a PPC task should be separated from the algorithmic solution to solve this task. This makes it not only clear for service consumers how to use this service but also allows use of the service without knowledge of the algorithmic solution, i.e. the implementation. Additionally, implementing PPC tasks as services motivates for a use-case independent implementation with a parameterized interface.

To be precise, a PPC task can be seen as a function f, where $f\colon\ x \rightarrow y$. Thereby, x is data required to perform the PPC task and y the solution of the PPC task. For example, data required for performing MRP comprises of the production schedule and the bill of materials of all products in the schedule and the output covers cardinalities for material. Considering the contextual meaning of both x and y respectively in f, it is possible to define their schemas X and Y. Schemas for MRP would be clear definitions of the data models describing the production schedule or the bill of materials. With this formulation, a PPC task can be implemented as a service where its interface conforms X and Y and its implementation realizes f.

With this logic, it is possible to transfer the logic of service composition and orchestration to PPC for creation of planning pipelines in PPC. For example, we could define a service that is a composite of scheduling and MRP, where the input of the composite service are the placed orders of customers and the associated bill of materials. Output of this service would be a production schedule and cardinalities for all required materials.

Thus, we could abstract the sequential PPC logic, i.e. performing MRP after scheduling, in this service composition and thereby reduce complexity. Considering the orchestration capabilities of SOA, complex process models for PPC services could be realized for automation.

3.2 Software architecture to implement service-orientation in manufacturing

To realize an integration of PPC services and production data sources, a software architecture is required that is able to deal with the data heterogeneity of manufacturing and allows to orchestrate complex service networks for PPC tasks. Aligning with the concept of SDM [22] and the architectures discussed in section 2, our proposed architecture consists of three layers - infrastructure layer, reference layer and application layer – as shown in Figure 1.

The infrastructure layer is thereby a collection of data providers and consumers, i.e. production hardware and data bases, and their associated way of communication. The reference layer serves two purposes: data integration and service integration. At first, it aims for integrating the infrastructure to a more manageable representation by use of data integration technologies and providing the infrastructure's functionalities as services. Next, service integration is concerned with the registration, integration and orchestration of services in order to make them discoverable, compatible, and manageable. Lastly, the application layer comprises of the services itself and the configuration interfaces to control the components of the architecture.

In the data integration of the reference layer, data heterogeneity needs to be handled considering notations and concepts of data. There have been great advancements in terms of reducing notational data heterogeneity in shopfloors by middleware communication technologies. One example is the Eclipse BaSyx middleware (https://www.eclipse.org/basyx/) that allows to connect multiple industry protocols and make the information available in AAS. However, other integrations might be necessary besides the shop floor to data bases or enterprise applications such as MES or ERP systems (see Section 3.3), that are considered by integration services. The AAS serves as a promising technology to realize the target of data integration of the reference layer by making the data available in a manageable representation. It is a standardized description language with a service-oriented design that suits well for standardization of data formats. Currently, there exist many ongoing standardization procedures to create distinct domain models for AAS to ease integration.

Apart from its advantages, AAS technology is complex to use, and, unfortunately, not yet as compatible and mature as existing technologies for SOA and web services. Realizing data models with AAS requires precise knowledge of the AAS meta model, posing a high barrier to entry. Although AAS allow for the creation of schemas using templates, existing implementations for AAS currently do not provide any validation of these schemas. Technologies that support building an SOA like load balancers, health monitoring or data integration tools for services are not compatible with AAS. Instead they are widely compatible to existing solutions for describing web services. One example, the OpenAPI Specification, is a definition language that allows to describe, produce, consume, and visualize web services in a machine-readable form.

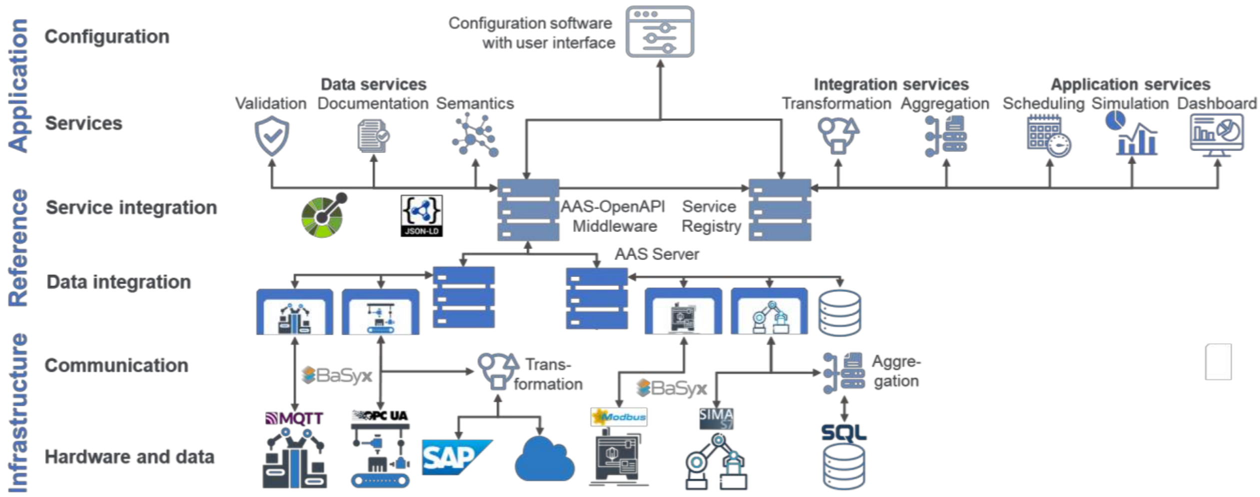

Figure 1: Visualization of the proposed architecture for realization of a service-oriented PPC

To make use of the benefits of AAS, i.e. standardized representation of data and compatibility to industrial data sources, and OpenAPI Specification, i.e. compatibility to SOA technologies and ease of use, we propose a middleware that integrates these languages. For this integration, we developed a mapping between the components of the meta models of AAS and OpenAPI schemas, as shown in Table 2. Note, that we considered only the most important components of both languages in this work. Extensions, however, to cover more aspects of the meta models are possible. Meta-information of schema and attribute names, necessary for this transformation, is specified within the DataSpecifications of the AAS. A more detailed explanation of this mapping based on a simple example can be found in the Appendix.

By considering these mapping rules, the AAS-OpenAPI middleware automatically transforms data between these two formats. To make the middleware useable, it provides a CRUD (create, read, update, and delete) REST API for its data models that is dynamically generated based on provided AAS or OpenAPI specifications. Queries against this CRUD interface are transformed and forwarded to the AAS and are transformed again upon return from the AAS interface. For an implementation of this middleware, refer to: [46]

Table 2: Mapping logic for transforming AAS to OpenAPI Specifications and vice versa

AAS Meta Model	OpenAPI	Mapping Transformation
Asset administration shell	object	ID, IDshort, and semantic ID are mapped as strings, submodels are mapped as attributes
Submodel	object	ID, IDshort, and semantic ID are mapped as strings, Submodel Element Collection are mapped as attributes
Submodel Element Collection	object	ID, IDshort, and semantic ID are mapped as strings, Submodel Element Collection are mapped as attributes
Submodel Element List	array	Values of the list entries are mapped to an array attribute
Property	string | number | integer | boolean	Value of the Property is mapped to the associated primitive data type attribute (either string, number, integer or boolean)
ReferenceElement	string	Value of the Reference Element is mapped as a string attribute

3.3 Data Integration and compatibility with legacy systems

The consideration of integration of legacy systems for PPC in this architecture is one essential component. The first step for integration is to ensure that these systems provide their functionalities as services, e.g. by wrapping their typical API in web services. Existing procedures are widely applicable and have been performed for many applications, such as in [30]. Although this allows to include legacy systems in the architecture, connecting them still requires data integration.

PPC tasks are strongly dependant on each other due to the sequential nature of PPC, input and output of different PPC tasks can have intersections or be subsets of each other. Let, for example, the output of scheduling service conform schema Y_S and let the input of an MRP service conform schema X_{MRP} that is an integration of a scheduling schema X_S and a bill of material schema X_{BOM}. Then, one would need to find a mapping M_S that integrates both scheduling schemas Y_S and X_S in order to integrate scheduling and MRP service.

Considering the number of tasks in PPC, the efforts for data integration with heterogenous interfaces can be huge. In fact, a worst-case scenario could require to define $\frac{1}{2}N(N-1)$ mappings for N schemas in a point-to-point integration [47]. However, concepts from data integration reduce this complexity by considering global or at least mediated schemas that reduce the number of necessary mappings in the best case N [31,29].

Semantically annotated data also has great potentials for data integration by use of schema matching technologies, automated semantic integration and ontology mapping [48,49]. Lastly, in case of missing semantics, one could employ existing machine learning approaches, or more specifically natural language processing, from other domains to automate the integration [50,51].

4. Conclusion and Outlook

This work aims to enable service-orientation in PPC by applying the principles of SOA to PPC and creating an architecture that allows the integration of such services with production data sources. At first, related approaches from literature are analyzed with consideration of their degree of service-orientation and their capabilities to handle heterogenous data sources. Analysis showed that the reviewed approaches either miss realization of some SOA principles or they rely on homogenous data sources. To close this deficit, we propose a concept that transfers theory from SOA and data integration to PPC frameworks, showing that PPC tasks can be described as services which are modular and composable. Based on this, an architecture is introduced that allows to synergistically use digital industry and SOA technologies for data exchange in production. It allows to orchestrate loosely coupled services to perform PPC tasks and to exchange data with production in an interoperable way. By integration of AAS with OpenAPI Specifications, the architecture achieves to handle notational data heterogeneity since the most common protocols and languages in practice can be interchangeably transformed. Lastly, special focus is given to handle legacy systems and conceptual data heterogeneity. Although the use of OpenAPI requires schema definitions of service interfaces, integrating these schemas is still linked to high efforts. To resolve this problem, integration with global schemas or use of automated approaches to find these schema mappings are recommended. In our future research, we will build up on this approach and demonstrate its effectiveness in use cases. Moreover, more detailed evaluation of the potentials of methods utilizing machine learning or semantics for data integration is focused.

Acknowledgements

The authors gratefully acknowledge financial funding from the German Federal Ministry of Economic Affairs and Climate Action and the European Union (grant no. 13IK001ZF) as well as organizational support by the VDI Technologiezentrum GmbH.

Appendix

In the following, we will explain the mapping defined in Table 2, for the integration of AAS and OpenAPI Specification schemas, based on a simple example. A reference implementation is available at [46]. The example is concerned with a data model (Figure 2a) that describes attributes of products that are required in typical PPC tasks, such as MRP or scheduling. The data model of the product specifies three attributes: an ID to identify the product, an attribute for required production processes and an attribute for its bill of material. The data model is depicted in Figure 2a as UML class diagram and an instance of the data model is shown in Figure 2b.

The result of applying the mapping logic between AAS and OpenAPI Specifications on this example is displayed in Figure 3. Here, the described data model is displayed in JSON-serialization for its representations conforming OpenAPI and AAS Meta model.

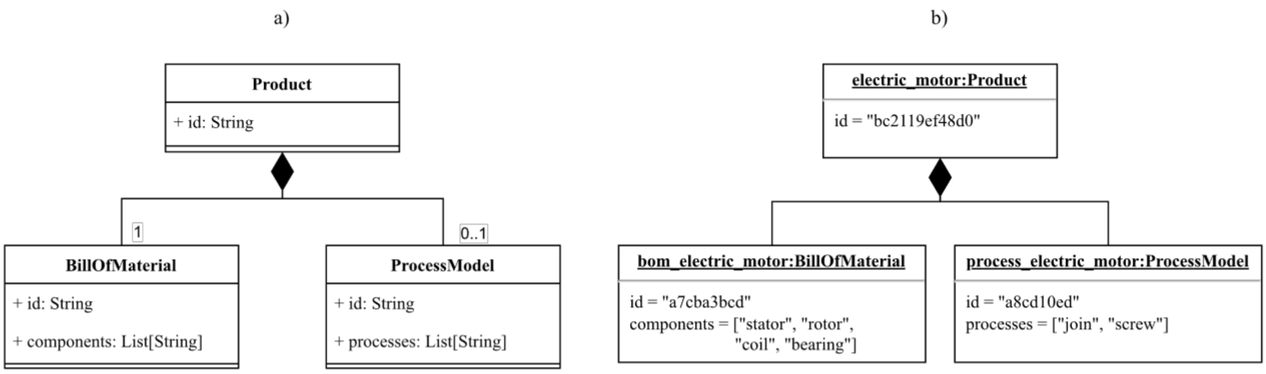

Figure 2: UML class diagram (a) and object diagram (b) of the exemplary data model of an electric motor

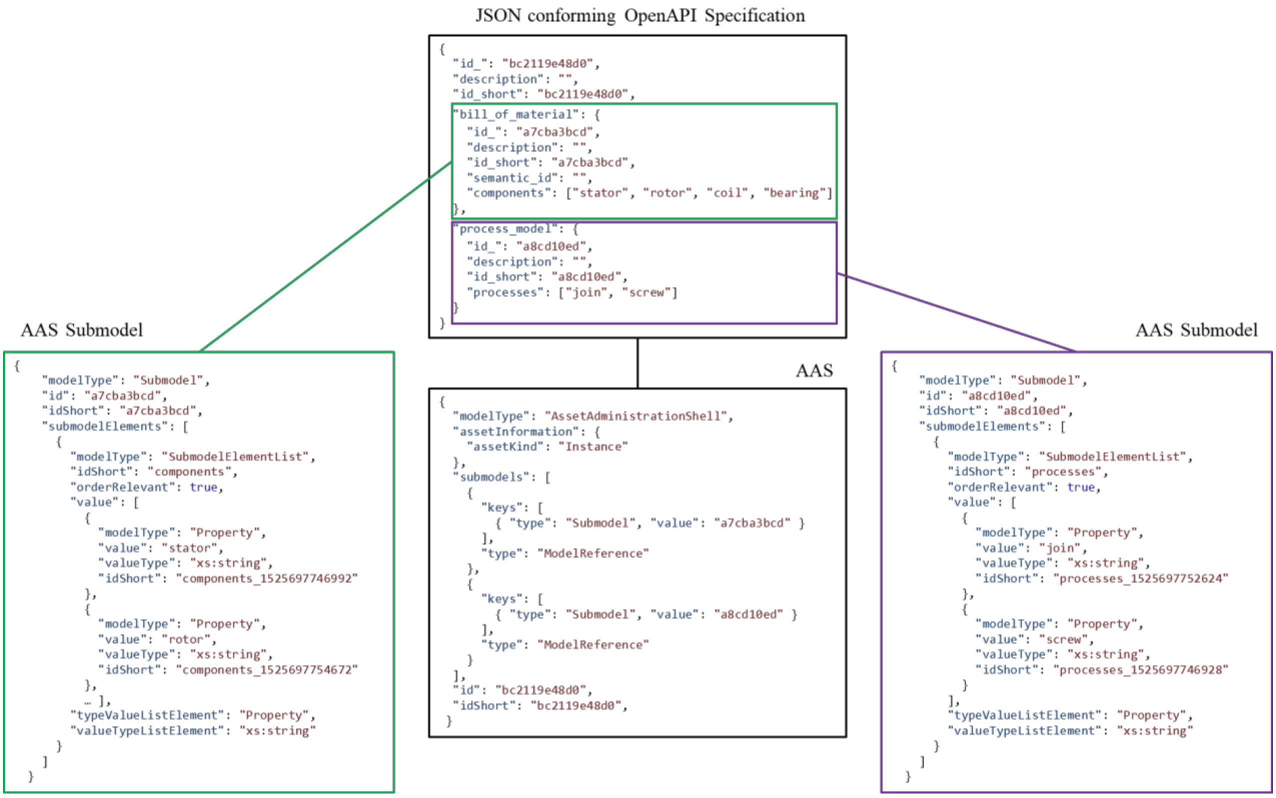

Figure 3: UML class diagram (a) and object diagram (b) of the exemplary data model of an electric motor

References

[1] Enke, J., Glass, R., Kreß, A., Hambach, J., Tisch, M., Metternich, J., 2018. Industrie 4.0 – Competencies for a modern production system: A curriculum for Learning Factories. Procedia Manufacturing 23, 267–272.

[2] Mack, O., Khare, A., Krämer, A., Burgartz, T. (Eds.), 2016. Managing in a VUCA world. Springer, Cham, Heidelberg, New York, Dordrecht, London, 259 pp.

[3] Millar, Carla C. J. M., Groth, O., Mahon, J.F., 2018. Management Innovation in a VUCA World: Challenges and Recommendations. California Management Review 61 (1), 5–14.

[4] Wiendahl, H.-P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.-H., Duffie, N., Brieke, M., 2007. Changeable Manufacturing - Classification, Design and Operation. CIRP Annals 56 (2), 783–809.

[5] Oluyisola, O.E., Sgarbossa, F., Strandhagen, J.O., 2020. Smart Production Planning and Control: Concept, Use-Cases and Sustainability Implications. Sustainability 12 (9), 3791.

[6] Bennett, N., Lemoine, G.J., 2014. What a difference a word makes: Understanding threats to performance in a VUCA world. Business Horizons 57 (3), 311–317.

[7] ElMaraghy, H.A., 2005. Flexible and reconfigurable manufacturing systems paradigms. Int J Flex Manuf Syst 17 (4), 261–276.

[8] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., van Brussel, H., 1999. Reconfigurable Manufacturing Systems. CIRP Annals 48 (2), 527–540.

[9] ElMaraghy, H., Wiendahl, H.-P., 2014. Changeable Manufacturing, in: Laperrière, L., Reinhart, G., Chatti, S., Tolio, T. (Eds.), CIRP encyclopedia of production engineering. With 85 tables. Springer, Berlin, pp. 157–163.

[10] Saputra, N., Sasanti, N., Alamsjah, F., Sadeli, F., 2022. Strategic role of digital capability on business agility during COVID-19 era. Procedia Computer Science 197, 326–335.

[11] Monostori, L., 2014. Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. Procedia CIRP 17, 9–13.

[12] Stark, R., Kind, S., Neumeyer, S., 2017. Innovations in digital modelling for next generation manufacturing system design. CIRP Annals 66 (1), 169–172.

[13] Overbeck, L., Le Louarn, A., Brützel, O., Stricker, N., Lanza, G., 2022. Comprehensive Validation Metrics and Precise Updating of Digital Twins of Production Systems. Simulation Notes Europe 32 (3), 135–142.

[14] Behrendt, S., Wurster, M., May, M.C., Lanza, G., 2023. Extended Production Planning of Reconfigurable Manufacturing Systems by Means of Simulation-based Optimization, in: 4th Conference on Production Systems and Logistics (CPSL 2023), Santiago de Querétaro, Mexiko. 2023. publish-Ing, Santiago de Querétaro, Mexiko, pp. 210–220.

[15] Kuhnle, A. Adaptive order dispatching based on reinforcement learning. Dissertation.

[16] Usuga Cadavid, J.P., Lamouri, S., Grabot, B., Pellerin, R., Fortin, A., 2020. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. Journal of Intelligent Manufacturing 31 (6), 1531–1558.

[17] Yang, C., Lan, S., Wang, L., Shen, W., Huang, G.G.Q., 2020. Big Data Driven Edge-Cloud Collaboration Architecture for Cloud Manufacturing: A Software Defined Perspective. IEEE Access 8, 45938–45950.

[18] Tseng, M.-L., Tran, T.P.T., Ha, H.M., Bui, T.-D., Lim, M.K., 2021. Sustainable industrial and operation engineering trends and challenges Toward Industry 4.0: a data driven analysis. Journal of Industrial and Production Engineering 38 (8), 581–598.

[19] Wiendahl, H.-P., Scholtissek, P., 1994. Management and Control of Complexity in Manufacturing. CIRP Annals 43 (2), 533–540.

[20] Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 51 (11), 1016–1022.

[21] Negri, E., Fumagalli, L., Macchi, M., 2017. A Review of the Roles of Digital Twin in CPS-based Production Systems. Procedia Manufacturing 11, 939–948.

[22] Behrendt, S., Ungen, M., Fisel, J., Hung, K.-C., May, M.-C., Leberle, U., Lanza, G., 2023. Improving Production System Flexibility and Changeability Through Software-Defined Manufacturing, in: , Cham. 2023. Springer International Publishing, pp. 705–716.

[23] Lechler, A., Verl, A., 2017. Software Defined Manufacturing Extends Cloud-Based Control, in: Proceedings of the ASME 12th International Manufacturing Science and Engineering Conference. ASME 2017 12th International Manufacturing Science and Engineering Conference collocated with the JSME/ASME 2017 6th International Conference on Materials and Processing, Los Angeles, California, USA. 6/4/2017 - 6/8/2017, New York, N.Y.

[24] Phuyal, S., Bista, D., Bista, R., 2020. Challenges, Opportunities and Future Directions of Smart Manufacturing: A State of Art Review. Sustainable Futures 2, 100023.

[25] Adamson, G., Wang, L., Holm, M., Moore, P., 2017. Cloud manufacturing – a critical review of recent development and future trends. International Journal of Computer Integrated Manufacturing 30 (4-5), 347–380.

[26] Niknejad, N., Ismail, W., Ghani, I., Nazari, B., Bahari, M., Hussin, Ab Razak Bin Che, 2020. Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation. Information Systems 91, 101491.

[27] Valipour, M.H., Amirzafari, B., Maleki, K.N., Daneshpour, N., 2009. A brief survey of software architecture concepts and service oriented architecture, in: . 2009 2nd IEEE International Conference on Computer Science and Information Technology. 2009/08, pp. 34–38.

[28] Mayring, P., 2015. Qualitative Content Analysis: Theoretical Background and Procedures, in: , Approaches to Qualitative Research in Mathematics Education. Springer, Dordrecht, pp. 365–380.

[29] Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., Belfkih, S., 2018. Big Data technologies: A survey. Journal of King Saud University - Computer and Information Sciences 30 (4), 431–448.

[30] Sneed, H.M., 2006. Integrating legacy software into a service oriented architecture, in: . Conference on Software Maintenance and Reengineering (CSMR'06). 2006/03, 11-pp.-14.

[31] Lenzerini, M., 2002. Data integration: a theoretical perspective, in: . SIGMOD/PODS02: International Conference on Management of Data and Symposium on Principles Database and Systems, Madison Wisconsin. 2002/06/03. ACM, pp. 233–246.

[32] Genesereth, M.R., Keller, A.M., Duschka, O.M., 1997. Infomaster: an information integration system. SIGMOD Rec. 26 (2), 539–542.

[33] Wang, B., Tao, F., Fang, X., Liu, C., Liu, Y., Freiheit, T., 2021. Smart Manufacturing and Intelligent Manufacturing: A Comparative Review. Engineering 7 (6), 738–757.

[34] Lee, J., Bagheri, B., Kao, H.-A., 2015. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters 3, 18–23.

[35] Pérez, F., Irisarri, E., Orive, D., Marcos, M., Estevez, E., 2015. A CPPS Architecture approach for Industry 4.0, in: . 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). 2015/09, pp. 1–4.

[36] Liu, C., Jiang, P., Jiang, W., 2020. Web-based digital twin modeling and remote control of cyber-physical production systems. Robotics and Computer-Integrated Manufacturing 64, 101956.

[37] Ye, X., Hong, S.H., Song, W.S., Kim, Y.C., Zhang, X., 2021. An Industry 4.0 Asset Administration Shell-Enabled Digital Solution for Robot-Based Manufacturing Systems. IEEE Access 9, 154448–154459.

[38] Wang, J., Xu, C., Zhang, J., Bao, J., Zhong, R., 2020. A collaborative architecture of the industrial internet platform for manufacturing systems. Robotics and Computer-Integrated Manufacturing 61, 101854.

[39] Biesinger, F., Meike, D., Kraß, B., Weyrich, M., 2019. A digital twin for production planning based on cyber-physical systems: A Case Study for a Cyber-Physical System-Based Creation of a Digital Twin. Procedia CIRP 79, 355–360.

[40] Qiu, R.G., 2007. A service-oriented integration framework for semiconductor manufacturing systems. International Journal of Manufacturing Technology and Management 10 (2-3), 177–191.

[41] Grassi, A., Guizzi, G., Santillo, L.C., Vespoli, S., 2020. A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. Manufacturing Letters 24, 43–46.

[42] Napoleone, A., Macchi, M., Pozzetti, A., 2020. A review on the characteristics of cyber-physical systems for the future smart factories. Journal of Manufacturing Systems 54, 305–335.

[43] Qu, Y.J., Ming, X.G., Liu, Z.W., Zhang, X.Y., Hou, Z.T., 2019. Smart manufacturing systems: state of the art and future trends. Int J Adv Manuf Technol 103 (9), 3751–3768.

[44] Trunzer, E., Calà, A., Leitão, P., Gepp, M., Kinghorst, J., Lüder, A., Schauerte, H., Reifferscheid, M., Vogel-Heuser, B., 2019. System architectures for Industrie 4.0 applications. Prod. Eng. Res. Devel. 13 (3), 247–257.

[45] Chapman, S.N., 2006. The Fundamentals of Production Planning and Control. Pearson/Prentice Hall.

[46] Sebastian Behrendt, 2023. aas2openapi v0.1.1, https://github.com/sdm4fzi/aas2openapi,. Zenodo, 10.5281/ZENODO.8209922.

[47] H. Mannaert, J. Verelst, Peter De Bruyn, 2016. Normalized systems theory : from foundations for evolvable software toward a general theory for evolvable design.

[48] Knoblock, C., Szekely, P., 2015. Exploiting Semantics for Big Data Integration. AI Magazine 36, 25–38.

[49] Mountantonakis, M., Tzitzikas, Y., 2019. Large-scale Semantic Integration of Linked Data: A Survey. ACM Comput. Surv. 52 (5), 103:1-103:40.

[50] Cappuzzo, R., Papotti, P., Thirumuruganathan, S., 2020. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks, in: , New York, NY, USA. 2020//31/Mai. Association for Computing Machinery, pp. 1335–1349.

[51] Zhang, J., Shin, B., Choi, J.D., Ho, J.C., 2021. SMAT: An attention-based deep learning solution to the automation of schema matching. Advances in databases and information systems. ADBIS 12843, 260–274.

Biography

Sebastian Behrendt (*1997) is a research associate at the Institute of Production Science (wbk) at the Karlsruhe Institute of Technology (KIT) and works in the group of production system planning. He studied mechanical engineering with a specialization in production systems, data analysis and machine learning at the Karlsruhe Institute of Technology (KIT).

Florian Stamer (*(1991) is a postdoctoral researcher and group leader at the Institute of Production Science (wbk) at the Karlsruhe Institute of Technology (KIT). He leads the Quality Assurance group, which focuses on intelligent quality control loops in production, with AI and advanced metrology as key enablers.

Marvin Carl May (*1994) is a researcher at the Institute for Production Science (wbk) at the Karlsruhe Institute of Technology (KIT). His research concerns production system optimisation on strategical, tactical and operational levels by means of intelligent algorithms. The focus lies on discrete manufacturing, in close cooperation with industry, e.g. semiconductor manufacturing, and aims at researching and implementing innovative and application-oriented solutions to technological problems.

Gisela Lanza (*1973) is member of the management board at the Institute of Production Science (wbk) of the Karlsruhe Institute of Technology (KIT). She heads the Production Systems division dealing with the topics of global production strategies, production system planning, and quality assurance in research and industrial practice. Her research focus is on the holistic design and evaluation of production systems