Automated Communication for Fault Diagnosis in Flexible Production Environments*

Pascal Rübel[1], Simon Jungbluth[2], William Motsch[2], and Martin Ruskowski[1]

[1] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
[2] Techologie-Initiative SmartFactory KL e.V., Kaiserslautern, Germany
pascal.ruebel@dfki.de

Abstract. Decentralised, modular production with the aim of individualised products leads to a more flexible production setup which, however, also influences the handling of faults and failures. Since faults occur rarely compared to nominal behavior of Cyber-Physical Production Modules (CPPM), it is difficult in common manufacturing environments and even harder in skill-based production to gain experience and knowledge about faults and the context they occur in. Hence, leveraging knowledge and data from multiple CPPM proves beneficial, facilitating the storage of acquired information regarding faults and their context in federated knowledge bases. However, although different approaches tackle the communication between distributed knowledge bases, the use for distributed knowledge-based fault detection and diagnosis in skill-based production environments remains mainly unseen. In this paper the focus lies on the development of a communication scheme that enables automated communication between fault detection and fault diagnosis components for a decentralised control setup to make distributed knowledge about faults accessible. This includes the definition of fault detection and fault diagnosis components and their offered services which encapsulate different forms of knowledge representations. For the communication between the components, a unified model is elaborated, and the required information is identified. An integration in a holonic manufacturing system of *SmartFactory*$^{\mathrm{KL}}$ is presented and an outlook for further research is given.

Keywords: fault diagnosis · skill-based production · agent-based-communication · holonic manufacturing systems

1 Introduction

More automation in fault detection and diagnosis (FDD) is a key factor to increase resilience and self-healing capabilities of manufacturing systems. However, the need for individualised products requires small batch sizes up to batch size

* This work was supported by the German Federal Ministry of Education and Research by the project "Rare Class Learning and Unknown Events Detection for Flexible Production" (RACKET) under grant 01IW20009.

one. Hence, more flexibility in factory automation is required, leading to modular factory setup and skill-based production [1, 2]. The concept of Cyber-Physical Production Systems (CPPS) enables flexibility due to multiple, interchangeable constellations of production systems but increases complexity, especially in handling of decentralisation of models and knowledge bases on the other hand. Even in common manufacturing environments it is a challenge to gain experience and knowledge about faults and failures since they occur rare when compared to nominal behavior. The increased complexity in skill-based CPPS makes it even more difficult. Individualised tasks and small lot sizes lead to small amounts of data and knowledge scaling up the challenge additionally.

One way to tackle the challenge is to leverage data and knowledge from multiple CPPM resulting in federated knowledge bases to store faults and their context. Knowledge bases about faults and their context are modelled in the Capability-Skill-Service-Fault-Symptom Model (CSSFS Model) in [3]. The model is transferred into a knowledge graph to make the knowledge accessible in a structured format. To increase availability, resilience and the autonomy level of CPPS, an automated decision making for FDD is required. However, even some approaches tackle the automated, industry-oriented communication between federated knowledge bases, the use case of FDD especially for skill-based production systems remains mainly open.

Against this background, the focus of this paper is to develop a communication scheme that enables automated communication between FDD components for a decentralised control setup, making distributed knowledge, also about faults, accessible. Therefore, the required FDD components and their tasks and services are defined, encapsulating distributed forms of knowledge. For each task, a heuristic communication scheme is defined to query the corresponding knowledge. The responses of the queries are the basis for autonomous or manual decision making. The scheme is implemented using a multi-agent system (MAS) at the real-world demonstrator at *SmartFactory*$^{\mathrm{KL}}$. This demonstrates the application of a Holonic Manufacturing System (HMS) to encapsulate the intricacies of systems comprising multiple individual subsystems essential to deal with the complexity in distributed skill-based production. Therefore, the entities responsible for FDD, along with their respective tasks, are individually defined and subsequently incorporated into the pre-existing holonic MAS that manages both direct and indirect manufacturing tasks for the CPPS demonstrator.

In Section 2, the current state of the art in skill-based production, HMS and fault diagnosis approaches, which are realized by MAS, is given. Required components for FDD in skill-based production including their tasks as well as a communication scheme between the components are outlined in Section 3. The integration of the components and the communication scheme into the real-world demonstrator of *SmartFactory*$^{\mathrm{KL}}$ and its MAS is shown with an example case in Section 4. Finally, the paper closes with a conclusion and future work in Section 5.

2 State of the Art

2.1 Modular, Skill-Based Production

Cyber-Physical Systems (CPS) play an important role in Industrie 4.0 [4]. A CPS can provide autonomous control and is crucial for the design of smart factories [5]. CPPMs are built of CPS and provide standardised interfaces for different functionalities [6]. Using these interfaces, CPPMs can be used to built CPPSs in the way of a flexible aggregation of lower-level components [7]. The Capability-Skill-Service Model is based on the PPR-model which provides a formalised and machine-readable description of manufacturing functions [8]. In a structure of CPPS, each CPPM can encapsulate and provide manufacturing skills, based on the concept of skill-based engineering [2]. Skills can be defined in the manufacturing context as the asset-dependent implementation of asset-independent capabilities [9, 10]. The relation of functionalities of skill-based engineering, considering the PPR-model, are also shown in [10]. There, capabilities are used to enable a product driven production. Product requirements are realised with more dynamic possible production processes, which manifest as skills on the shop floor. To follow the paradigm of a Shared Production, production skills can furthermore be abstracted and provided as decentralised production services [11].

2.2 Holons in the Context of Skill-based Production

The skill-based approach enables the connection between the OT and the IT layer by standardised interfaces with self-contained functionality. Reconfiguration of the production lines is possible by dynamic parametrisation and skill sequencing. For this purpose intelligent system behavior is needed to deal with disturbances efficiently by using the skill interface.

A manufacturing paradigm to deal with this challenge is named HMS. This system consists of one or multiple holons that can be simultaneously a part and a whole. It represents an analogy of a system consisting of several autonomous subsystems [12]. A more technical definition describes a holon as an autonomous and co-operative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. The holon consists of an information processing part and often a physical processing part [13]. These concepts are manifested in the reference architecture PROSA [14] that covers aspects of hierarchical and heterarchical control architectures. In PROSA product, order and resource holons are responsible for one aspect of manufacturing control each. The basis of PROSA served as inspiration for the development of new architectures, e.g., ADACOR. ADACOR [15] defines four holons: product, task, operational and supervisor holon. The aim was to balance between a more centralised approach and a flatter one. In case of machine failure, the hierarchical architecture reorganises heterarchical to achieve an alternate plan. ARTI [16] refined and improved PROSA by avoiding the usage of manufacturing specific wordings. The authors distinguish between digital twins and decision-making elements. In this context, resource type, resource instance, activity type and activity instance are introduced as elements of the world of interest. The boundary is defined by digital twins for decision-making. These digital replicas, operating in a non-physical realm, facilitate the virtual execution of decision makers' intentions on corresponding resource instances faster than real time.

The concepts HMS, digital twins and also CPPS have great potential for combination. This is demonstrated in our approach in [17], inspired by PROSA, ARTI and the CSS-Model [8, 14, 16]. Holons are combined with standardised interfaces of the skill-based approach and associated digital twins known as Asset Administration Shell (AAS). Service, product and resource holons are defined to handle intra- and inter-organisational optimisation. However, at this stage, it is exemplary shown how to solve planning and scheduling issues to ensure production dynamically. Currently malfunctioning resources are not considered any further. In this work, the extensions of the CSSFS will be added to dynamically detect and diagnose faults [3].

2.3 Multi-Agent based Fault Diagnosis

HMS are mainly motivated by the manufacturing domain, following the goal of increasing flexibility. Approaches in the field of fault diagnosis are primarily located in the field of engineering and more and more in artificial intelligence. The role of agents is to demonstrate the collaboration of different entities incorporating different tasks each. Therefore, in case of fault diagnosis in HMS the tasks of fault detection and of fault diagnosis need to be covered and knowledge from different sources need to be used to elaborate a holonic solution. In this sub-section the related work in the field of multi-agent based fault diagnosis is shown.

The field of online safety monitoring covers the tasks of FDD, alarm annunciation and fault controlling. A hierarchical MAS covering different system levels and a distributed monitoring model is used to develop the safety monitor. The monitoring model is derived from a safety assessment model and consists of a behavior model of hierarchical state machines and a fault propagation model of multiple fault trees. The MAS is a set of Belief-Desire-Intention Agents that are able to reason locally at sub-system level and at global system level using a collaboration protocol. [18–21] Li, Wang & Wu introduce a distributed fault diagnosis system reference model based on MAS covering the tasks of detection of local and global anomal behavior, fault analysis and diagnosis, causes identification, preventive and corrective maintenance tasks running as well as maintenance planning and the development of predictive models. To tackle the challenge, agents coordinate the interactions and are able to execute reasoning, adaptive and corrective aspects in their behaviors. The agents have access to various data from different sources to fulfill their tasks. In this paper the following agents are introduced taking over different tasks: Detect Agent, Analysis Agent, Diagnosis Agent, Manage Agent, User Agent and DB Agent. [22]

Mendoza, Xu & Song develop a fault diagnosis multi-agent model for petrochemical plants. The concept of leadership is introduced that combines the benefits of centralised and decentralised coordination schemes. Multiple agents monitor a set of sensors in the plant for normal and anomal states. The first agent that detects an anomal state takes the leader role and aggregates state information from the other agents. The leader agent builds a spatio-temporal pattern of anomal events that is compared to existing fault templates to identify the fault. [23, 24] A multi-agent architecture for distributed fault diagnosis problems is developed in [25]. The framework is split into knowledge abstraction, data acquisition, fault diagnosis inference and a user interface. Four agent types are introduced: Supervisor Agent, HCI Agent, Facilitator Agent and Diagnostic Agents. The Diagnostic Agents are responsible for fault detection and isolation in a local domain part of the system. A Hidden Markov Model is shown that evaluates the results of the local fault detection, thus, enabling the isolation of faults.

Although different approaches tackle the challenge of automated fault diagnosis, the distributed character of flexible manufacturing systems is rarely focused. Additionally, the integration of fault diagnosis task in a MAS to handle skill-based control tasks of a manufacturing system to make a further step towards autonomous self healing of distributed systems remains unseen.

3 Architecture for Automated Communication to Enable Fault Diagnosis in Skill-Based Production Environments

FDD is a key in complex manufacturing systems to foster a high level of resilience. Fast detection and advanced diagnosis of faults are the basis for high productivity and intelligent decision making in faulty scenarios. Holonic MAS can be used for encapsulated and modularised control of production modules and to establish communication and access to required information sources for FDD. A schema for an automated communication between fault detection and diagnosis components using distributed knowledge bases is needed and must be considered in an overarching concept. Therefore, the fault detection and diagnosis components as well as their tasks are described. Communication between the components and their decision making process with their related message contents and a heuristic communication scheme are presented.

3.1 Overview communication schema

Different functionalities of multiple factory layers can be encapsulated using holonic MAS. For a resilient production, a FDD is needed that requires three components to be implemented: a fault detection component, a fault diagnosis component and a knowledge base. In general, the fault detection component triggers a request for a diagnosis tasks in case of a deviation from nominal behavior. Therefore, it needs to submit the required information to the fault diagnosis component that queries the knowledge base for potential solutions (see Fig. 1).

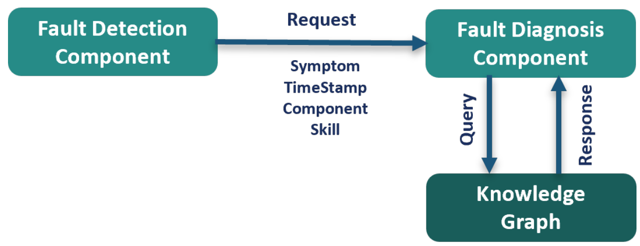

Fig. 1. Overview of the communication schema

3.2 Fault Detection Component

In general, the task of Fault Detection is to determine whether a fault is present in a system or not. Hence, in skill-based production, we tried to determine faults on a resource during the execution of one or multiple skills. The determination of faults rely on the generation of symptoms that referred to a specific fault class. For the generation of these symptoms, it is possible to use phenomenological or model-based approaches, whereas phenomenological approaches use classifiers to model the direct relationship between input variables and the symptoms. Model-based approaches model the nominal behavior of a system that is compared with the actual behavior. If there is a significant deviation between nominal and actual behavior corresponding symptoms are generated. Following the work of [26] for fault detection in skill-based production, model-based approaches are chosen. Therefore, the introduction of the tasks of monitoring, nominal modelling and symptom generation was needed.
The monitoring component provides observations of a defined set of features that are characteristic to describe the behavior of a system. The same set of features is modelled to describe the nominal behavior of the same system. Finally, those values are compared and symptoms are generated if they deviate significantly. Further information on the tasks can be found in [26] and are not further elaborated here since the focus of this paper is the corresponding communication scheme.
The calculated symptoms initiate the communication with the diagnosis component. Required output for the following diagnosis tasks are the elaborated set of symptoms, the resource of the current fault as well as the skill that was executed during the occurrence of the fault and a timestamp to be able to use information from other sources. Resource and skill elements are submitted for identification purposes, so static identifier are sufficient there. The submission is mandatory. Contrasting to this, the length of the set of symptoms needs to be variable since before the calculation it is not clear how many symptoms are observable in the system. To be able to detect a fault, at least one observable symptom is required.

3.3 Fault Diagnosis Component

The task of the Fault Diagnosis Component is to analyse characteristics of a fault based on the inputs of the Fault Detection Component. This can include various sub-tasks that determine kind, size, location or causes and the severity of faults. In this paper, the focus is on tasks for classification, root cause analysis and recommendation for fault handling. For the task execution a heuristic query scheme is followed whereas the tasks are executed in the aforementioned sequence, since the inputs of the previous tasks are required at each step. Additional tasks can be introduced and put at the required spot in the sequence, if they are necessary.
In this work, a knowledge-based approach for fault diagnosis was used. The characteristics of a fault, as well as its context, are modelled in an information model and made accessible using knowledge graphs. Each fault was modelled at least with its symptoms, the resource it occurred on, the product that was produced and the skill that was executed during the occurrence of the fault. This follows the CSSFS-Model developed in [3].
After modelling, the knowledge graph needed to be queried to get access to the stored knowledge used for answering the fault diagnosis tasks. Each task requires an own definition for the used queries. Since there are different scenarios possible in each task, a hierarchical query scheme for each task was developed.
The heuristic communication scheme queries different scenarios iterative, starting with the one having the most precise and easiest answer. If no match can be found, the next step of the scheme is executed. Once all steps are executed without having a match, the fault is classified as unknown. In that case, a human expert needs to analyse the situation and define it in the knowledge base.

For the classification task, the query in Step 1 tries to find an exact match of fault symptoms on the same resource, executing the same skill. If a match exists the scheme the can be exited and reply the fault class that was found. In Step 2, similar faults on the same resource are looked up if there was no match in Step 1. This is the case when just a subset of the corresponding set of symptoms are observed during the execution of the same skill or the same set of symptoms during the execution of another skill. The match of Step 2 is a fault class suggestion because there was no exact match for all queried input parameters. Step 3 addresses similar faults on other resources, but on the same resource type, for example a milling machine of the same vendor. Different definitions of similarity of resources can be done and can be concreted domain specific and use case specific. In Step 4, similar faults on other resources and other resource types are looked up. Additional similarities of the resource can be taken into account, like same components or same functions to make sense of the given results. The query scheme is shown in Fig. 2. Finally, if none of these steps deliver a response, the inputs from the Fault Detection Component are classified as unknown. After a human analysis of the system, the fault definition can be added to the knowledge base.

The other tasks can be modelled analogous and are not displayed.

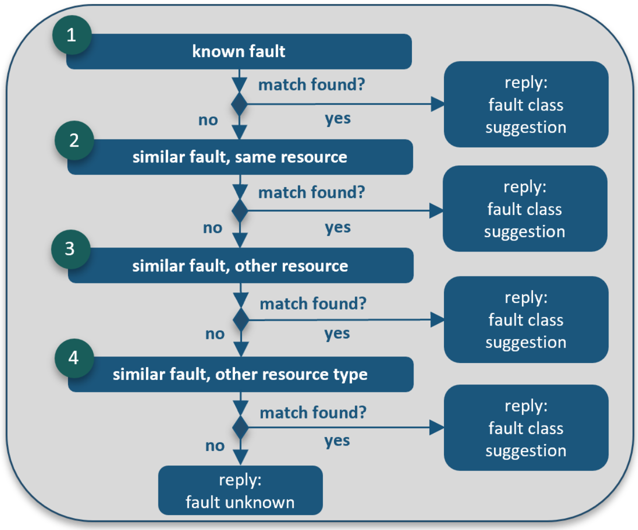

Fig. 2. Heuristic communication scheme for a fault classification task

4 Implementation and Results

The integration in this work is shown on a CPPS of the *SmartFactory*$^{\mathrm{KL}}$ production ecosystem. The CPPS consists of several CPPMs, providing different skills to be able to produce product in small batch sizes (see Fig. 3).

For the fault detection component the nominal behavior was modelled with physical models on atomic skill level and suitable sensors are attached for monitoring the features. For the fault diagnosis component the faults and their context were modelled using AAS. Faults itself, the resources and skills were modelled in resource AAS and product information in a product AAS. The AAS were transformed in Neo4j label property graphs and follow the information model elaborated in [3] to be accessible.
The Shared Production HMS of *SmartFactory*$^{\mathrm{KL}}$ consists of three holons: Service Holon, Product Holon and Resource Holon. For self description purposes each holon has its own AAS. An Identification Submodel serves for identification and a submodel for the interfaces of the holon to enable communication using different technologies. A topology submodel describes the internal structure of the holon consisting of several sub-holons. The Service Holon provides and manages the services of the factory to the world outside of the factory. The Product Holon divides the required tasks for the production into sub-tasks and handles the production process. Finally, the Resource Holon covers the management of the CPPMs using different APIs like OPC UA and are connected to the AAS. Resource Holons of the type CPPS and CPPM are differentiated, whereas a CPPS Holon demonstrates the existence of multiple sub-holons and the CPPM Holon abstracts the smallest possible entity that is not further partitioned. On an inter-

Fig. 3. CPPS of *SmartFactory*$^{\mathrm{KL}}$

action level, this differentiation has no effect. CPPS Holons cover lifecycle management and coordination tasks and behaviors, whereas CPPM Holons perform and schedule tasks on a concrete level. Additionally to AAS, Lifecycle and I4.0 Message Skills, the CPPS Holon offers the following behaviors: Update, Inter-Holon, Negotiation and Human Behavior. Especially the Human Behavior needs to be emphasised because human expertise is required for improvement of the modelled knowledge in the graph and for the case of previously unknown faults. On the CPPM Level AAS, I4.0 Message Skill and Human Behavior was included as well. Furthermore, Bidding and OPC UA Skills and Requirement Check, Bidding, Neighbor, Execution and Monitoring Behavior are covered. More detailed descriptions of the Skills and Behaviors can be found in [17].
Following the description in Section 3, FDD components with their tasks needed to be integrated into the existing HMS to create a holonic approach that can handle the whole manufacturing. Since existing Behaviors and Skill allowing the communication and monitoring of the CPPM on a control and hardware level, the fault detection component was integrated into the Resource Holon on the CPPM Level. Therefore, the already existing Monitoring Behavior was adapted and a Behavior simulating nominal behavior of the CPPM and a behavior for generating symptoms were added to the portfolio.
The fault diagnosis components require more coordination capabilities since the knowledge bases of multiple CPPMs are required to find similar cases if no match is found on the CPPM the fault occurred on. Accordingly, the fault diagnosis component was attached to the CPPS Holon. A visualisation of the Resource Holon and the integration of FDD is shown in Fig 4.

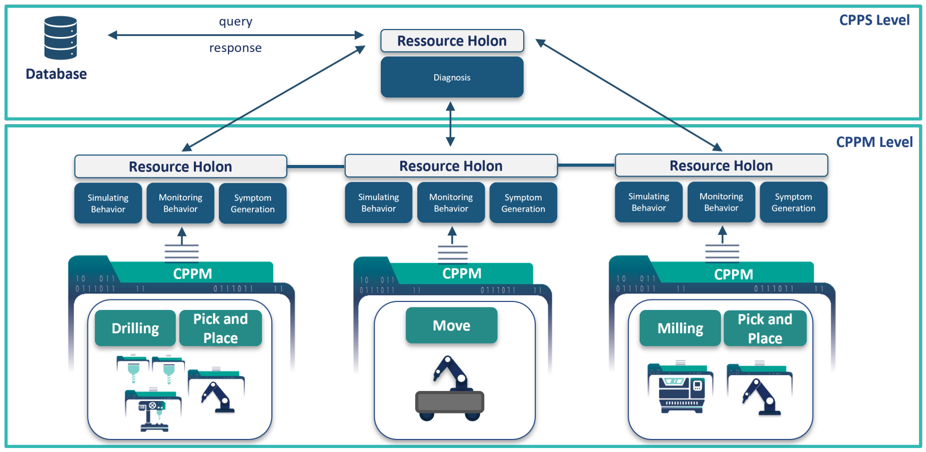

Fig. 4. Visualisation of Resource Holon with integrated FDD

5 Conclusion & Outlook

In this paper a communication scheme for integrated and autonomous FDD for skill-based production environments is developed to make distributed control setups and knowledge bases accessible. Therefore, required components for FDD and their tasks are elaborated and a communication scheme between the components is outlined. The fault detection components include the tasks of monitoring, nominal behavior and symptom generation. Once a deviation between nominal and actual behavior is present, symptoms, timestamps, skill and resource information are submitted to the fault diagnosis component and an analysis is triggered. The tasks of fault diagnosis subsume e.g. classification, root cause analysis and handling recommendations. Additional tasks can be introduced when they are required and integrated accordingly.

To elaborate the tasks, the fault diagnosis component queries a knowledge-base that stores knowledge about faults and their manufacturing context using a heuristic query scheme. The scheme allows to efficiently access the knowledge instead of running resource intensive similarity analysis in the first step. The steps start with a direct full match and use other scenarios that are queried, if there is none. Once there is no match found, the fault is declared as unknown and a human expert is required to analyse the situation and define the fault then in the knowledge base.

Finally, the components and interaction scheme are integrated and tested into the existing HMS of *SmartFactory*$^{\mathrm{KL}}$. With the integration a crucial component can be added to the HMS to reach the goal of a resilient and almost autonomous HMS, that provides a decision support to human workers. The elaborated FDD components better suit the requirements of skill-based production system. Rather than using anomaly detection or monitoring techniques, the nominal behavior is simulated that is compared to actual behavior. Additionally, the knowledge-based approach for diagnosis allows more possibilities for analysis. Due to this assumptions from previous work, that differs from other approaches, the architecture for the components and the communication followed. In contrast to the leadership concept introduced in [23] the coordination is handed to the diagnosis component and does not stay with the first holon, since the diagnosis component manages the sequence of the tasks as well as the scheme within the tasks. The diagnosis component is integrated on the CPPS level of the existing HMS, so it can undertake the communication with other holonic agents. This was out of scope for this work, but needs to be tackled in the future. Since the scope of this work was on the communication scheme, future work can focus on the messages between the components. The structure of messages based on the I4.0 specification can be included. Additionally, further diagnostic tasks and similarity approaches can be added in future work.

References

1. Hermann, J., Rübel, P., Birtel M., Mohr, F., Wagner, A., Ruskowski, M.: Self-description of cyber-physical production modules for a product driven manufacturing system. In: Procedia Manufacturing, 38, pp. 291–298 (2019).
2. Ruskowski, M. et al: Production Bots für Production Level 4: Skillbasierte Systeme für die Produktion der Zukunft. In: atp magazin, 62 (9), pp. 62-71 (2020)
3. Rübel, P., Moarefvand, N., Motsch, W., Wagner. A., Ruskowski, M.: Enabling Fault Diagnosis in Skill-Based Production Environments. In: Proceedings of the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), Sinaia, Romania, 2023, pp. 1-8 (2023)
4. Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative Industrie 4.0: Final report of the Industrie 4.0 Working Group. Forschungsunion: Berlin, Germany.
5. Ribeiro, L. (2020). System design and implementation principles for industry 4.0-development of cyber-physical production systems. Studentlitteratur AB, Lund, 3, 10-15.
6. Kolberg, D., Hermann, J., Mohr, F., Bertelsmeier, F., Engler, F., Franken, R., ... & Staub-Lang, P. (2018). SmartFactoryKL System Architecture for Industrie 4.0 Production Plants. SmartFactoryKL, Whitepaper SF-1.2, 4.
7. Birtel, M., Mohr, F., Hermann, J., Bertram, P., & Ruskowski, M. (2018). Requirements for a human-centered condition monitoring in modular production environments. IFAC-PapersOnLine, 51(11), 909-914.
8. Köcher, A. et al: A reference model for common understanding of capabilities and skills in manufacturing. In: at - Automatisierungstechnik, 71 (2), pp. 94-104 (2023)
9. Diedrich, C., Belyaev, A., Bock, J., Grimm, S., Hermann, J., Klausmann, T., ... & Winter, M. (2022). Information Model for Capabilities, Skills & Services.
10. Motsch, W., Dorofeev, K., Gerber, K., Knoch, S., David, A., & Ruskowski, M. (2021, September). Concept for modeling and usage of functionally described capabilities and skills. In 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-8).
11. Bergweiler, S., Hamm, S., Hermann, J., Plociennik, C., Ruskowski, M., & Wagner, A. (2022). Production Level 4-Der Weg zur zukunftssicheren und verlässlichen Produktion. Whitepaper SF-5.1.

12. Van Brussel, H.: Holonic manufacturing systems the vision matching problem. In: First European Conference on Holonic Manufacturing Systems, Hannover, Germany. 1994.
13. Van Leeuwen, E. H., Norrie, D.: Holons and holarchies. In: Manufacturing Engineer, vol. 76 (2), pp. 86-88 (1997)
14. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L, Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. In: Computers in industry, vol. 37 (3), pp. 255-274 (1998)
15. Leitão, P., Colombo, A. W., Restivo, F. J.: ADACOR: A collaborative production automation and control architecture. In: IEEE Intelligent Systems, vol. 20 (1), pp. 58-66 (2005)
16. Valckenaers, P.: Perspective on holonic manufacturing systems: PROSA becomes ARTI. In: Computers in Industry, vol. 120, p. 103226 (2020).
17. Bernhard, A., Jungbluth, S., Karnoub, A., Sidorenko, A, Motsch, W., Wagner, W., Ruskowski, M., I4.0 Holonic Multi-Agent Testbed Enabling Shared Production, Artificial Intelligence in Manufacturing, (2024), to be published
18. Dheedan, A., Papadopoulos, Y.: Multi-Agent Monitor. In: IFAC Proceedings, vol. 10 (4), pp. 84-89 (2010)
19. Dheedan, A., Papadopoulos, Y.: Model-Based Distributed On-line Safety Monitoring. In: ERMERGING 2011: The Third International Conference on Emerging Network Intelligence, pp. 1-7 (2011)
20. Dheedan, A.: Multi-agent On-line Monitoring for the Safety of Critical Systems. In: International Journal of Computer and Information Engineering, vol. 6 (11), pp. 1501-1511 (2012)
21. Dheedan, A.: On-line Safety Monitor Based on a Safety Assessment Model and Hierarchical Deployment of a Multi-agent System. In: International Journal on Advances in Internet Technology, vol 5 (3 & 4), pp. 95-113 (2012)
22. Li, Y., Wang, Hairui, Wu, L.: Multi-Agent based Design for Distributed Fault Diagnosis Systems. In: Advanced Materials Research, vol. 179-180, pp. 1266-1271 (2011)
23. Mendoza, B., Xu, P., Song, L.: A multi-agent model for fault diagnosis in petrochemical plants. In: IEEE Sensors Applications Symposium, pp. 203-208 (2011)
24. Mendoza, B., Xu, P., Song, L.: A Multi-agent Model with Dynamic Leadership for Fault Diagnosis in Chemical Plants. In: Abraham, A., Corchado, J.M., González, S.R., De Paz Santana, J.F. (eds) International Symposium on Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing, vol 91., pp. 19-26, Springer, Berlin, Heidelberg.
25. Ren, X., Thompson. H.A., Fleming, P.J.: An agent-based system for distributed fault diagnosis. In: International Journal of Knowledge-based and Intelligent Engineering Systems, vol. 10, pp. 319-335 (2006).
26. Rübel, P., Motsch, W., Schäfer, H., Ruskowski, M.: On Bringing Fault Detection to Skill-Based Production. In: Proceedings of the 1st European Symposium on Artificial Intelligence in Manufacturing (ESAIM), accepted, Kaiserslautern, Germany (2023)