ESAIM 2024 – 2nd European Symposium on Artificial Intelligence in Manufacturing

Agent-Based Communication for Fault Diagnosis in Skill-based Production Environments Using Messages based on I4.0 Language and Asset Administration Shells

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

[1] Technologie-Initiative SmartFactory KL e.V., Kaiserslautern, Germany
[2] German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

[*] Corresponding author. Tel.: +49-631-205751077;. E-mail: pascal.ruebel@smartfactory.de

Abstract. Small batch sizes and individualized products are managed with the concept of Cyber-Physical Production Systems (CPPS), enabling flexibility through interchangeable constellations, but increases complexity, especially when dealing with dependencies between decentralized subsystems. To increase the resilience and self-healing capabilities, greater automation of fault detection and diagnosis (FDD) is a key factor. It is a challenge to gather knowledge about faults, as these rarely occur compared to normal behavior. The flexibility in skill-based production systems makes this situation even more difficult. To overcome this challenge, data and knowledge about faults and their context from several Cyber-Physical Production Modules is used, which leads to federated knowledge databases. The knowledge databases are modeled in the Capability-Skill-Service-Fault-Symptom model (CSSFS model). To achieve the goal of high availability, resilience and autonomy of CPPS, automated decision-making for FDD using CSSFS applications is required. Therefore, automatic communication between FDD components is necessary. Therefore, focus of this paper is on the development of a communication scheme, which models participants using Asset Administration Shells and the I4.0 Language to model their interactions to enable automated communication and makes distributed knowledge accessible. To ensure decentralized control of these services, functionalities from several factory levels are encapsulated by Multi-Agent Systems (MAS) that follow a holonic structure.

Keywords: Fault Diagnosis · Holonic Manufacturing Systems · I4.0 Language · Asset Administration Shell.

1 Introduction

Greater automation of FDD is a key factor in increasing the resilience and self-healing capabilities of manufacturing systems. However, the demand for individualized products requires small batch sizes down to batch size one, that leverages the challenge of FDD. One solution to increase flexibility in factory automation is a modular factory structure and skill-based production. The concept of Cyber-Physical Production Systems (CPPS) enables flexibility through multiple, interchangeable constellations of production subsystems, but on the other hand increases complexity, especially when dealing with the decentralization of models and knowledge bases[1,2]. Even in non-decentralized production environments, it is a challenge to gather experience and knowledge about so-called faults and failures, as these rarely occur compared to the normal behavior of such systems. The increased flexibility in skill-based CPPS makes this situation even more difficult. Individualized tasks and small batch sizes lead to small amounts of individual data and knowledge, which extends the challenge of overall data usage.

One way of overcoming this challenge is to use data and knowledge from several Cyber-Physical Production Modules (CPPM), which lead to federated knowledge databases in which faults are stored with their context. Knowledge databases about faults and their context are modeled in the Capability-Skill-Service-Fault-Symptom model (CSSFS model).

In order to achieve the goal of high availability, resilience and autonomy of CPPS, automated decision-making for FDD using CSSFS applications is required. This requires a communication option that is as autonomous as possible between the components required for FDD.

Against this background, the research objective of this work is on the development of a communication scheme that enables automated communication between FDD components for a decentralized control structure and makes distributed knowledge accessible. Therefore, on the one hand the topic of Asset Administration Shell (AAS) for modelling of CPPMs as communication participants is focused. On the other hand the modelling of the messages between the participants using Industrie 4.0 language is examined. For this purpose, the required FDD components and their tasks and services are defined, which encapsulate distributed forms of knowledge. In order to ensure decentralized control of these services, various functionalities from several factory levels are encapsulated by MAS, that follow a holonic structure. For each task, a heuristic communication scheme is defined to retrieve the corresponding knowledge.

The paper is structured as follows. Section 2 gives an overview of related work in the field. In Section 3 an architecture describes the modelling of communication participants and the messages between them using I4.0 components and AAS as well as I4.0 Language. A prototypical implementation of the concept is shown in Section 4, whereas Section 5 discusses the results and gives an outlook for further research.

2 State of the Art

Following the introduction a brief state of the art in the identified main topics of modular and skill-based production, AAS and I4.0 language as well as fault diagnosis based on multi-agent systems is given in this section.

2.1 Modular, Skill-Based Production

A main element of a modular and flexible production environment is the topic of CPPM, which can provide standardized interfaces for their functionalities and be combined to build CPPS [3]. The CPPMs thus encapsulate their functionality by using the skill-based approach for usage in a flexible production structure [4]. Production skills are part of the information model for capabilities, skills and services (CSS-Model), in which capabilities are described as abstracted functions that are required in a production process and skills as the implementation of such a function and deployed on a specific production resource [5]. Skills can be implemented with industry standards such as OPC UA, to realize a generic interface that can be accessed and executed on higher control levels of a production system [6].

Skills of production resources are useable in a distributed organized control software, enabling their application into an agent-based setup within a MAS [7]. The integration of production skills of production modules into a MAS using resource agents and the design of skills for agent functionality is shown in [8]. In the current contribution, the fault diagnosis in skill-based production environments using agent-based communication is focused.

2.2 Asset Administration Shell and I4.0 Language

The Asset Administration Shell (AAS) is a standard provided by the I4.0 platform and standardized by the Industrial Digital Twin Association with the aim to implement a vendor-independent Digital Twin [9]. In the context of this article, AASs are used as a standardized interface between different subsystems of the CPPS for data exchange, as proposed in [10]. Thereby, AASs extend and represent production assets and leverage them in that way to I4.0 components.

These I4.0 components are accessed by an agent as a part of a holonic multi-agent system (MAS) [11]. This approach uses the agent definition of [12], in which agents are autonomous, problem-solving and goal-driven entities, observing and acting upon an open and dynamic environment. AASs configure and parameterize these agents and represent a standardized interface enabling a heterarchical communication between agents, which enables agents to act and react dynamically to changes in the environment. By the usage of holonic MAS, in the sense that agents can access, dynamically spawn and kill other agents along the ISA-95 factory hierarchy, the AASs of the respective I4.0 components are also hierarchically structured.

Accordingly, the AAS can be used to describe a standardizable structure of interfaces that can also be linked with semantic technologies. For example, the use of knowledge graphs and RDF stores offers potential for applications in the context of Industry 4.0, as presented in our approach in the following sections.

Complete AASs and AAS elements are sent through messages in the I4.0 language. VDI/VDE 2193 standardizes this message format. This guideline is divided into two parts. The first part [13] describes the vocabulary and structure of messages, and the second [14] the semantic interaction protocols. An I4.0 message consists of a message frame and the message content defined by AAS elements [9,10].

First, the frame contains several required elements: the message type, the AAS ID of the sender, the message ID, and a semantic protocol depending on the message type. Specifically, the message type defines the intention of the message, e.g., data inquiries, while the AAS ID and message ID ensure unambiguous tracking and order integrity. In addition, the semantic protocol describes the standardization of message intentions. Secondly, the message content is specified by the interaction elements, for example, the referenced AAS element.[9,10]

The interaction protocol of the I4.0 language is defined through a bidding procedure in which external participants communicate with internal participants (AAS sender/recipient) that communicate in the I4.0 language. This communication is supported by an I4.0 language handler, that transforms data elements into the I4.0 message format.[9,10]

2.3 Fault Diagnosis based on Multi-Agent Systems

In the field of fault diagnosis in manufacturing, the role of MAS is the incorporation of different entities that cover different tasks each. The use of such MAS increases the flexibility in encapsulating multiple resources, tasks and knowledge sources that enable an automated FDD in flexible production systems. For the task of fault diagnosis that is encapsulated in a holon, that is in this work realized by a MAS, a knowledge-based approach is used. The related work in the fields of FDD based on MAS and knowledge based FDD is shown in the following subsection.

For the automation of a fault diagnosis system in power systems, McArthur et al. use a combination of MAS and intelligent systems. Different intelligent decision support systems based on SCADA data and fault records are wrapped within agents to enable automated communication between the systems [15].

In [16], the KARMEN MAS is introduced that covers the task of process monitoring and notification. Component Agents are used that provide operational data of each process component, as well as Condition Monitor Agents that contain logical expressions to evaluate the conditions of the components. The logical expressions are manually defined by the user and can include data of multiple components. After evaluation of the Condition Monitor Agent, a Notification Agent handles the notification and escalation process [16].

The concept of leadership in MAS brings advantages of centralized and decentralized architectures together. Therefore, a set of agents observe sensor values and evaluate them for normal and anomal behaviour. The leader agent is the agent that firstly detects the anomal behaviour. The information of the other agents are aggregated and a spatio-temporal pattern is generated. The patterns are compared with existing fault patterns for fault identification [17,18].

The safety monitor is realized with a hierarchical MAS including a federated monitoring model and handles multiple system levels. Agents locally reason at subsystem level and collaborate for global reasoning. For monitoring, a model of hierarchical state machines is used for behaviour modelling. This model is combined with a fault propagation model, that consists of numerous fault trees [19,20,21,22].

A hybrid reinforcement learning MAS is proposed in [23] focusing the identification of anomalies in industrial microservices. Local outlier determination is executed by agents for each microservice using reinforcement learning. Afterwards the local results are merged to extract global outliers by an intelligent communication strategy.

In the area of knowledge-based fault diagnosis a semantic framework, that describes the manufacturing domain is introduced in [24]. Manufacturing resources, processes and their context are used in combination with system observations to enable stream observations to diagnose situations potentially leading to failures [24]. In [25] connection and component models are developed, that calculate fault symptoms based on components behaviour and their relations. Once a faulty component is identified, the system is reconfigured to bypass it. The new configuration is then fed back as a system update [16]. A general overview of the state of the art of fault diagnosis on CPPS is given by Niggemann and Lohweg in [26].

Although, the stated work in this subsection covers fault diagnosis in CPPS it still lacks the integration of the CSS-model and AAS for knowledge representation and the combination with holonic MAS for automated communication to enable fault diagnosis. Therefore, in this work the focus is on the CSSFS-Model presented in [27] and its usage as a basis for knowledge representation and communication schema introduced in [28] for automated communication. The communication schema is extended with modelling its participants as actice AAS and the message based on I4.0 messages paving the way to higher interoperability, resilience and autonomy of the concept.

3 Structure for Automated Communication for Fault Diagnosis in Flexible Production Systems

For the integration of FDD in flexible production systems, three components are required to be implemented on suitable hierarchy levels of CPPS: A fault detection component, a fault diagnosis component and a knowledge base.

In general, the Fault Detection component triggers a request for a diagnostic task in the event of a deviation from the target behavior. It must therefore transmit the required information to the fault diagnosis component, which queries the knowledge base for possible solutions.

The main task of the fault detection component is the determination of one or multiple faults present in the system. Therefore, a set of symptoms is generated that relates to a fault class. A detailed description of the applied fault detection task can be found in [29]. The fault diagnosis component uses the input of the fault detection component to elaborate analysis on fault classification, root cause analysis and recommendations on the handling of faults. Accordingly, a heuristic communication scheme for the FDD components to execute the different tasks is presented in [28].

In this work, the focus is in the modelling of the communication participants as I4.0 components based on AAS and the messages on I4.0 language (see section 2.2) to enable a knowledge-based approach for fault diagnosis. The properties of a fault and its context are modeled in an information model and made accessible via knowledge graphs. Each fault is modeled at least with its symptoms, the resource on which it occurred, the product produced and the skill that was executed when the fault occurred. This follows the concept of the CSSFS model [27]. Both, the production context and the faults themselves are initially modelled in corresponding AASs to have the knowledge available in a standardized, structured format. To increase searchability and accessibility, these AASs are transformed into knowledge graphs. Doing this, the I4.0 components representation can be used for communication purposes and the knowledge graph representation for analysis and searchability purposes.

In the following, the I4.0 components representation of the communication participants are described. Afterwards, the message structure is described based on the I4.0 language standard.

3.1 Communication Participants

For the usage of I4.0 language, the communication participants need to be modelled as I4.0 components. Therefore, the physical asset is extended with a digital representation to act as a I4.0 component. In this work, type 2 AAS are used as a form of digital representation. AAS of type 2 are reactive AAS, that can be used as an interface. I4.0 components are modelled on the CPPM and CPPS level, whereas multiple CPPM belong to a corresponding CPPS.

On both hierarchy levels submodels are added that contain general information about the asset that are not FDD related in the first place. The mentioned submodels are a minimal configuration and can be extended, depending on the domain and use case. For the FDD use case the modelled information can be used for further analysis.

A digital nameplate submodel, that contains information as usually found on physical nameplates in an interoperable manner, is added to the AAS. Additionally, a submodel for the Bill of Material structure is introduced. This submodel contains the hierarchical structure of industrial equipment that can be composed of different subsystem levels. On the CPPM level, the components of the CPPM are described, whereas on the CPPS level the topology of the corresponding CPPM is modelled. Each CPPS subsystem can have its own AAS. The information about the hierarchical resource structure is required for FDD tasks. Finally, the manufacturing skills that a system can provide is modelled in a submodel. There, the required information about each offered skill is modelled separately in

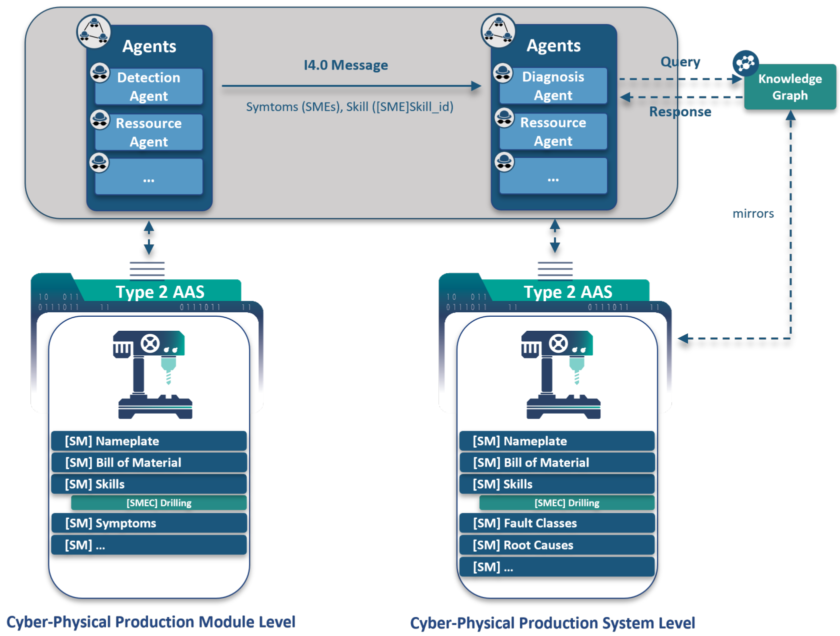

Fig. 1. Overview of the system including participants and messages

a submodel element collection. The information includes an ID for identification purposes and properties for parametrization of the skill.

Additionally to the AAS modelling of the physical components the tasks of FDD need to be integrated in the CPPS hierarchy. The Fault Detection component with its task to monitor executed skills, simulate nominal behavior and generation of symptoms is integrable in a MAS on the CPPM level. A detailed description of the Fault Detection tasks can be found in [29]. Consequently, a submodel is added containing the information about the symptoms that are generated by the symptom generation task. Since the fault detection, and accordingly the symptom generation, is executed on the CPPM level, the symptom submodel extends the CPPM AAS.

Based on the introduced concepts, an integration of a detection agent into a MAS can be proposed, that executes the corresponding tasks and triggers communication with the other agents on CPPM and CPPS level.

Depending on the tasks of the Fault Diagnosis component, further submodels need to be added to extend the information of the AAS. Since information of all CPPMs of the CPPS is available on the CPPS level, the diagnosis is located there. For the tasks of fault classification and root cause determination, a submodel containing fault classes and root causes can both be added. Analog to the CPPM level, a Diagnosis Agent can be suggested for integration in a MAS that handles the tasks and communication.

For a better accessibility and searchability, the AAS are transformed in a knowledge graph, like described in [27], that can be accessed by the Diagnosis Agent. The use of a knowledge graph representation ensures a higher accessibility and enables more sophisticated analysis like similarity analysis. The system overview is shown in Figure 1.

3.2 Message Structure

After modeling faults and their context, the knowledge graph must be queried to access the stored knowledge that is used to answer the fault diagnosis tasks. Each task requires a separate definition for the queries used. As different scenarios are possible, a hierarchical query schema is developed for each task.

Both, the communication between the fault detection and the fault diagnosis component and the access between the fault diagnosis component and the knowledge graph are modeled according to the I4.0 language.

Using the I4.0 language full AAS or AAS elements can be sent in a standardized format. I4.0 messages consist of a message frame and content, that needs to be defined by a list of AAS elements.

In the frame message, type and ID as well as the AAS ID of the sender are mandatory. Since the message type defines the intention of the message, the message type “fault diagnosis” is used. The AAS ID of the sender is the AAS ID of the CPPM that triggers the interaction with the Diagnosis Agent. The message ID is automatically set and ensures unambiguous tracking and tracing of the messages.

The content of the message is specified by interaction elements in form of AAS or AAS elements. In the FDD use case the symptoms that have been detected as well as information about the resource and the executed skill need to be submitted.

Therefore, the submodel element collections of the generated symptom as well as the submodel element containing the skill ID are added to the message content. The required information about the resource is already part of the message frame in the form of the AAS ID of the sender.

4 Proof of Concept and Results

The system will be implemented as a holonic MAS in the *SmartFactory*$^{\mathrm{KL}}$ demonstrator environment. This demonstrates the application of a holonic manufacturing system to encapsulate the intricacies of systems consisting of multiple individual subsystems that are essential for managing complexity in distributed capability-based manufacturing.

Therefore, the entities responsible for FDD along with their respective tasks are defined individually and then integrated into the already existing holonic MAS. Both direct and indirect manufacturing tasks are managed for the CPPS demonstrator, which can ensure a holistic view of the integration of the FDD components in the manufacturing context.

The AAS were modelled using the Eclipse AASX Package Explorer™ [1] and then deployed using Eclipse BaSyx middleware [2]. For the implementation of the holonic MAS in this work SARL language running on Janus Agent and Holonic Platform [3] is used. The communication between the agents follows the standardized structure of Industrie 4.0 language.

4.1 Agent communication for fault detection and fault diagnosis

The functionality for fault detection and diagnosis is encapsulated as own agents in the prototypically implemented MAS of the *SmartFactory*$^{\mathrm{KL}}$ and realized with the Janus Framework, which is an open-source framework for the development of holonic agents, based on the agent-oriented programming language SARL [30]. The agent communication is shown in Figure 2.

The relevant communication part in the system begins with a request for a production skill execution by a resource agent, which controls the skill execution of a production module. The resource agent informs the fault detection agent about a scheduled skill execution on this production module to prepare a skill monitoring. The fault detection agent monitors the behavior of corresponding production modules during the skill execution and considers the result of this monitoring activity. Detected fault-related data are prepared by the fault detection agent for further analysis by the fault diagnosis agent. The request for fault diagnosis is then used in the fault diagnosis agent to prepare a request to the knowledge base, in which the diagnosis is processed. As a knowledge base, a knowledge graph is used as presented in [27], so that the fault diagnosis agent encapsulates and provides the functionality to build a query for accessing the stored information of this knowledge graph.

4.2 Asset Administration Shell and Agent based Structure and Communication Implementation

For the implementation, one AAS for the Resource Agent of the running production module and one AAS for the *SmartFactory*$^{\mathrm{KL}}$ demonstrator are applied. Both in Figure 3 displayed AASs are modelled and implemented in AAS version 3.

The CPPM agent AAS contains all described standardized submodels such as Nameplate, BOM, Skills, Technical Data, Asset Interface Description, Monitoring, and Symptoms. During the execution of the production module, the detection agent detects new symptoms and adds them to the AAS. As soon as the execution stops, aborts or gracefully ends with a monitoring result, the detection agent sends all newly detected symptoms together with the respective skill ID of the agent to the central diagnosis agent to request a fault diagnosis. The diagnosis agent then checks the detected symptoms via the knowledge

[1] https://github.com/admin-shell/aasx-package-explorer
[2] https://eclipse.dev/basyx/about/
[3] http://www.sarl.io/runtime/janus/

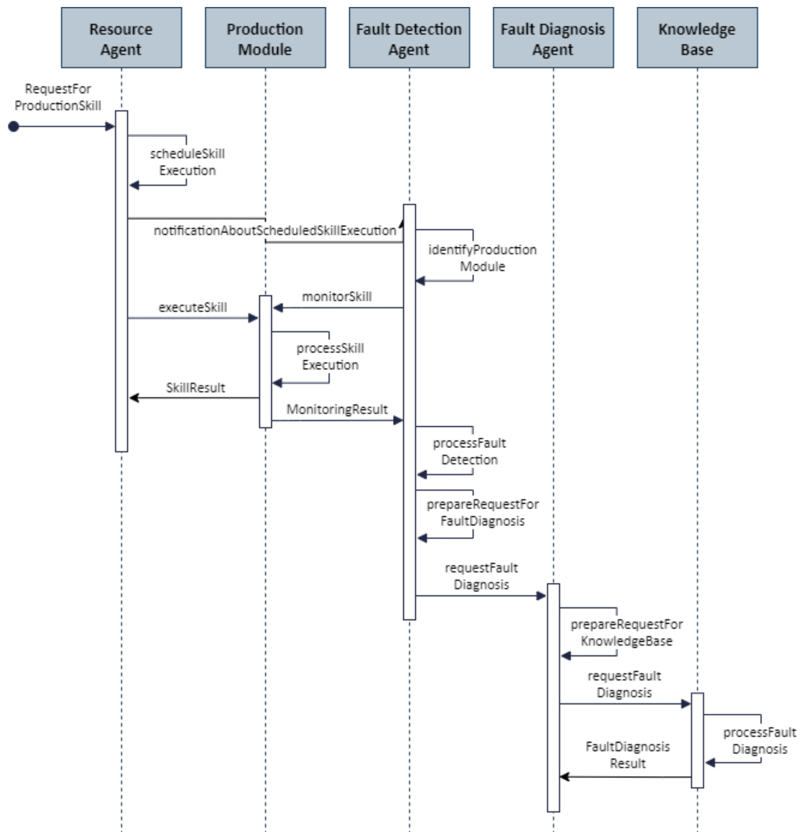

Fig. 2. Sequence Diagram for Agent Communication for Fault Detection and Fault Diagnosis

graph, like describe in [27], and adds references to known faults stored on the CPPS AAS to the symptoms submodel on the production module AAS.

For communication between different agents, messages are sent via interaction protocols. For this purpose, all agents of the MAS have the ability to communicate event-based in the same event space inside the MAS, which is provided by the Janus framework for agent communication. This works perfectly for agents running on the same or closely connected event spaces. This is the case for the CPPM agent and the detection agent, since the CPPM agent dynamically spawns the detection agent on starting a production task and both agents thus share the same agent context. The same situation appears between the CPPS and the diagnosis agent. However, the detection agent and the diagnosis agent might not be deployed on the same device and take part in the same event space. This is why all messages between the agents are embedded into an industry 4.0 message framework, adding more meta-information. Based on the specification

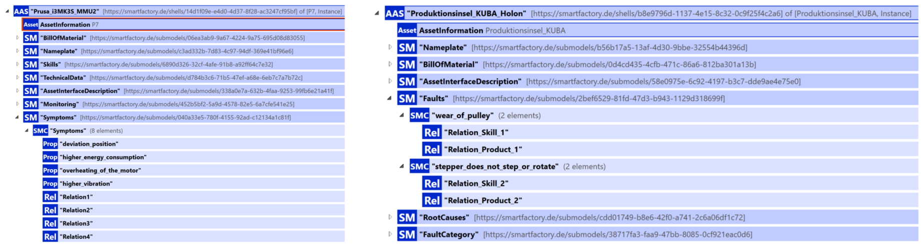

Fig. 3. Asset Administration Shells of a CPPM agent (left side) and the CPPS agent (right side)

of the I4.0 message in [12], a metamodel for I4.0 Messages is defined in Java which contains all mandatory and optional fields of I4.0 messages.

5 Discussion and Conclusion

An agent-based communication structure for the integration of FDD in skill-based production environments is elaborated. The involved participants are described and modelled using type 2 AAS whereas the communication is realized by a MAS and a message structure on the basis of the I4.0 language. Type 2 AAS and the corresponding agents act together similar to the concept of the active type 3 AAS.

In the first step, the FDD tasks were integrated into the different hierarchy levels of CPPS. The Fault Detection Tasks are executed on the CPPM level due to the monitoring and simulation tasks of the behaviour of the CPPM during the execution of manufacturing skills. The tasks are modelled as agent behaviours and encapsulated in a Fault Detection Agent, that covers the communication with other participants. The CPPM as well as the fault symptoms are modelled as AAS and AAS submodels.

The Fault Diagnosis tasks are integrated on the CPPS level, since the access to the information of the whole CPPS is available there. The CPPS is modelled as type 2 AAS as well and required submodels for fault diagnosis are added. A Fault Diagnosis Agent is introduced that manages the internal elaboration of the requested tasks and the communication with participating agents.

The message structure is based on the I4.0 Language standard defined in [9,10]. Therefore, the needed information and its AAS representation is defined and integrated in the message structure.

Finally, the defined structure is prototypically implemented as part of the holonic MAS of the *SmartFactory*$^{\mathrm{KL}}$ demonstrator environments. Besides the proof of concept the basic interactions between the existing agents in the MAS are shown. In this work, the focus was on the detection and diagnosis of a priori known faults. In this case, it is possible to get existing matches of instances, that are already in the knowledge base. In the case of a priori unknown faults, further analysis needs to be executed to elaborate suggestions on possible fault classes and root causes. Future research can tackle the topic of similarity analysis for suggestions on a priori unknown fault cases. Machine Learning techniques, especially graph learning, can be used for the similarity analysis as well as graph completion in future work.

Additionally, the scenario of the usage of knowledge across company borders is especially interesting in the FDD case, since faults in industrial environments occur rare compared to nominal behaviour. The challenge of sharing data and knowledge of multiple companies using data spaces can be developed in future research.

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. Kolberg, D., Hermann, J., Mohr, F., Bertelsmeier, F., Engler, F., Franken, R., Kiradjiev, P., Pfeifer, M., Richter, D., Salleem, M.: SmartFactoryKL System Architecture for Industrie 4.0 Production Plants; Whitepaper SF-1.2, 4; SmartFactoryKL: Kaiserslautern, Germany, 2018.
4. Bergweiler, S.; Hamm, S., Hermann, J., Plociennik, C., Ruskowski, M., Wagner, A.: Production Level 4-Der Weg zur Zukunftssicheren und Verlässlichen Produktion; Whitepaper SF-5.1; SmartFactoryKL: Kaiserslautern, Germany, 2022.
5. Diedrich, C., Belyaev, A., Bock, J., Grimm, S., Hermann, J., Klausmann, T., Köcher, A., Meixner, K., Peschke, J., Schleipen, M.: Information Model for Capabilities, Skills & Services; Fraunhofer-Gesellschaft: München, Germany, 2022.
6. Dorofeev, K., Zoitl, A.: Skill-based engineering approach using opc ua programs. In 2018 IEEE 16th international conference on industrial informatics (INDIN), pp. 1098-1103, IEEE, 2018.
7. Ruskowski, M., Herget, A., Hermann, J., Motsch, W., Pahlevannejad, P., Sidorenko, A., Bergweiler, S., David, A., Plociennik, C., Popper, J.: Production Bots für Production Level 4: Skill-basierte Systeme für die Produktion der Zukunft. Atp Mag., 62, 62–71, 2020.
8. Sidorenko, A., Motsch, W., Van Bekkum, M., Nikolakis, N., Alexopoulos, K., Wagner, A. (2023). The MAS4AI framework for human-centered agile and smart manufacturing. Frontiers in Artificial Intelligence, 6, 2023.
9. Federal Ministry for Economic Affairs, and Energy (BMWi), Industrial Digital Twin Association: Details of the Asset Administration Shell Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 2.0.1), `https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V2.pdf?__blob=publicationFile&v=6`, 2019.
10. Federal Ministry for Economic Affairs, and Energy (BMWi), Industrial Digital Twin Association: Details of the Asset Administration Shell Part 2 - Interoperability at Runtime – Exchanging Information via Application Programming Interfaces (Version 1.0RC02),`https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.pdf?__blob=publicationFile&v=1`, 2021.

11. Jungbluth, S., Hermann, J., Motsch, W., Pourjafarian, M., Sidorenko, A., Volkmann, M., Zoltner, K., Plociennik, C., Ruskowski, M.: Dynamic replanning using multi-agent systems and asset administration shells. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). S. 1–8. IEEE, 2022.
12. Leitão, P., Karnouskos, S.: Industrial agents: emerging applications of software agents in industry. Elsevier, 2015.
13. VDI/VDE 2193 Blatt 1 - Sprache für I4.0-Komponenten - Struktur von Nachrichten. 2020.
14. VDI/VDE 2193 Blatt 2 - Sprache für I4.0-Komponenten - Interaktionsprotokoll für Ausschreibungsverfahren. 2020.
15. McArthur, S.D.J., Davidson, E.M., Hossack, J.A., McDonald, J.R.: Automating power system fault diagnosis through multi-agent system technology. In: 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the. S. 8 pp. IEEE, Big Island, HI, USA, 2004.
16. Bunch, L., Breedy, M., Bradshaw, J.M., Carvalho, M., Suri, N., Uszok, A., Hansen, J., Pechoucek, M., Marik, V.: Software agents for process monitoring and notification. In: Proceedings of the 2004 ACM symposium on Applied computing. S. 94–100. ACM, Nicosia Cyprus, 2004.
17. Mendoza, B., Xu, P., Song, L.: A multi-agent model for fault diagnosis in petrochemical plants. In: 2011 IEEE Sensors Applications Symposium. S. 203–208. IEEE, San Antonio, TX, USA, 2011.
18. 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., und De Paz Santana, J.F. (hrsg.) International Symposium on Distributed Computing and Artificial Intelligence. S. 19–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
19. Dheedan, A.A.: Multi-Agent On-Line Monitor For The Safety Of Critical Systems, 2012. https://doi.org/10.5281/ZENODO.1075613
20. Dheedan, A.A.: Distributed on-line safety monitor based on safety assessment model and multi-agent system, 2012.
21. Dheedan, A., Papadopoulos, Y.: Multi-Agent Safety Monitor. IFAC Proceedings Volumes. 43, 84–89, 2010. https://doi.org/10.3182/20100701-2-PT-4011.00016
22. Dheedan, A., Papadopoulos, Y.: Model-based distributed on-line safety monitoring. In: The Third International Conference on Emerging Network Intelligence (EMERGING 2011), Lisbon, Portugal. S. 1–7, 2011.
23. Belhadi, A., Djenouri, Y., Srivastava, G., Lin, J.C.-W.: Reinforcement learning multi-agent system for faults diagnosis of mircoservices in industrial settings. Computer Communications. 177, 213–219, 2021. https://doi.org/10.1016/j.comcom.2021.07.010
24. Giustozzi, F., Saunier, J., Zanni-Merk, C.: A semantic framework for condition monitoring in Industry 4.0 based on evolving knowledge bases. SW. 15, 583–611, 2024. https://doi.org/10.3233/SW-233481
25. Diedrich, A., Balzereit, K., Niggemann, O.: First Approaches to Automatically Diagnose and Reconfigure Hybrid Cyber-Physical Systems. In: Beyerer, J., Maier, A., und Niggemann, O. (hrsg.) Machine Learning for Cyber Physical Systems. S. 113–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2021.
26. Niggemann, O., Lohweg, V.: On the diagnosis of cyber-physical production systems. State-of-the-art and research agenda, 2015. https://doi.org/10.24406/PUBLICA-FHG-388129

27. Rübel, P., Moarefvand, N., Motsch, W., Wagner, A., Ruskowski, M.: Enabling Fault Diagnosis in Skill-Based Production Environments. In: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). S. 1–8. IEEE, Sinaia, Romania, 2023.
28. Rübel, P., Jungbluth, S., Motsch, W., Ruskowski, M.: Automated Communication for Fault Diagnosis in Flexible Production Environments. Proceedings of the Flexible Automation and Intelligent Manufacturing international conference, accepted. Taichung, Taiwan, (2024)
29. Rübel, P., Motsch, W., Schäfer, H., Ruskowski, M.: On Bringing Fault Detection to Skill-Based Production. Proceedings of the 1st European Symposium on Artificial Intelligence in Manufacturing (ESAIM), accepted. Kaiserslautern, Germany, (2023)
30. Rodriguez, S.; Gaud, N.; Galland, S. SARL: A general-purpose agent-oriented programming language. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Warsaw, Poland, 11–14 August 2014; IEEE: Piscataway, NJ, USA, 2014; Volume 3, pp. 103–110.