FTOnto: A Domain Ontology for a Fischertechnik Simulation Production Factory by Reusing Existing Ontologies

Patrick Klein, Lukas Malburg, and Ralph Bergmann

Business Information Systems II, University of Trier, 54286 Trier, Germany
{kleinp,malburgl,bergmann}@uni-trier.de
http://www.wi2.uni-trier.de

Abstract Nowadays, semantic information provided by an ontology is indispensable in the context of Industry 4.0, especially when using methods from Artificial Intelligence. The currently available ontologies do not satisfy the demands of simulation environments used for research purposes. For this reason, we develop an ontology customized to Fischertechnik simulation factories by reusing existing ontologies. The ontology has been created according to requirements from two use cases. In our evaluation, it is determined that the ontology is suitable to represent machine components and their relationships while satisfying the specified requirements.

Keywords: Ontology Engineering · Industry 4.0 · Simulation Factory · Fischertechnik

1 Introduction

In recent years, the industry has been in the process of changing towards the fourth industrial revolution, also known as Industry 4.0 in the German-speaking area [10,14]. This transformation is characterized by manufacturing and service innovations based on *Cyber-Physical Systems (CPSs)*, big data, and the predominant use of *Artificial Intelligence (AI)* methods [16]. Although a lot of research work is carried out in this area today, there is still a lack of companies in the industry that are willing to make sensor or machine data available for research purposes or allow direct intervention in productive systems. As a consequence, research data must be generated by using appropriate simulation environments. In our previous work [11], we use a *Fischertechnik (FT)* simulation production factory. Such simulation factories are often used in research, e.g., for augmented reality [12], to plan, create, and evaluate different factory layouts [13], for research of digital twins by Fraunhofer IESE[1] or other related work (e.g., [22,30])

[1] https://www.iese.fraunhofer.de/de/presse/current_releases/PM_2019_02_25_Hannover-Messe.html, accessed May 31, 2019.

to name just a few. However, there are no suitable ontologies available to use them for simulation environments, although ontologies are a part of knowledge modeling and represent important knowledge. Furthermore, they are also needed to apply methods from AI such as case-based reasoning [1] or advanced machine learning. Current ontologies used such as the *Manufacturing's Semantics Ontology (MASON)* [17] and *Manufacturing Service Description Language (MSDL)* [2] do not consider sensor data streams and the *CREMA Data Model, Core module (CDM-Core)* [21] ontology is too comprehensive and detailed with classes and properties that are not needed in the context of a simulation environment. This paper addresses this issue and thus the development of a domain ontology for simulation factories. Therefore, it is investigated which components can be adopted from existing, related ontologies and which additional components need to be added. In the following, Sect. 2 introduces foundations for our work and discusses related work. Section 3 describes the layout of our used Fischertechnik simulation factory and presents use cases in which the application of semantic information provided by an ontology could be important. The development process and the developed ontology itself is described in detail and evaluated in Sect. 4. Finally, a conclusion is given and future work is discussed in Sect. 5.

2 Foundations and Related Work

Ontologies are used to describe the knowledge about a domain of interest in a formal way that can be understood by machines [9]. Therefore, the *Web Ontology Language (OWL)* provides classes, properties, individuals, and data values to express ontologies [29], which means that complex knowledge about individuals, groups of individuals, and their relationships can be represented [8]. Using an ontology benefits knowledge sharing between computational entities, knowledge reuse by using well-defined domain ontologies as well as the application of logical reasoning [19]. Moreover, an ontology forms the basis for the further use in knowledge-intensive applications [5].

A survey of upper ontologies regarding their modeling capabilities of a manufacturing system in terms of *products*, *processes*, and *resources* is recently conducted by Cao et al. [4]. One of the surveyed ontologies is the upper OWL-ontology *MASON* [17] that conceptualizes the manufacturing domain with three concepts: *entities*, *operations*, and *resources*. Similarly, the OWL-DL *Manufacturing Service Description Language (MSDL)* ontology [2] focuses on the manufacturing process as the core class of manufacturing services. The *Manufacturing System Engineering (MSE)* [18] ontology focuses on the inclusion of different taxonomies of teams in a manufacturing enterprise to improve collaboration rather than modeling the manufacturing process itself. These ontologies suffer from poor modeling of sensor data streams, which is an essential aspect of Industry 4.0. The latest and according to the authors largest publicly available ontology to model production and maintenance is the OWL2 *CREMA Data Model, Core module (CDM-Core)* [21], which is among other things an extension of *MASON* and the *Semantic Sensor Network (SSN)* ontology [7] to describe their data streams as well as their observed data and to provide an example for condition monitoring.

3 Fischertechnik Simulation Production Factory Model

Since a lot of information can be extracted from data of a real manufacturing environment (e.g., production quantity, processing times, failures, etc.), there are serious confidentiality concerns that make it often impossible for universities to obtain data for research. In addition to data, knowledge must also be available in order to semantically model the corresponding relationships of individual components. For this reason, simulated data is obtained from a model factory for research purposes. This section describes the layout of the factory and two exemplary use cases.

3.1 Layout of the Factory

For the simulation of an Industry 4.0 manufacturing environment, we use the *Fischertechnik (FT)* factory model shown in Fig. 1.

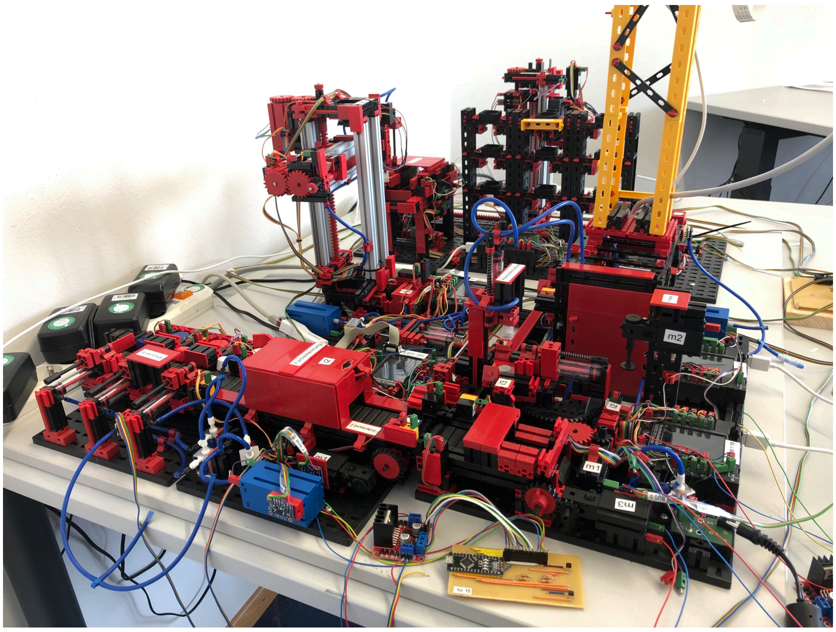

Figure 1. The FT Factory Simulation Model

It consists of four workstations with five individual modules: a sorting line with color detection, a multi-processing workstation with an oven and a milling machine, a high-bay warehouse, and a vacuum gripper robot. Each module is operated by its own controller based on an ARM Cortex A8 CPU with various analog and digital input/output ports running under a LINUX kernel. The model is equipped with nine light barriers and ten switches for control purposes of the actuators consisting of ten motors, three compressors, and eight valves. For condition monitoring purposes, the model is enhanced with dedicated sensors such as four three-axis acceleration sensors that are mounted on motors and compressors and four differential pressure sensors measuring the pressure generated from the three compressors. Furthermore, two absolute orientation sensors, each with a gyroscope, an accelerometer, and a geomagnetic sensor are installed on the robotic vacuum gripper and the dispensing machine of the high-bay warehouse. Similar to the continuous transformation of a factory in the context of Industry 4.0, the model is in a continuous development phase so that in future more components such as RFID reader/writers as well as additional processing and transport units will be integrated.

Each module of the factory is steered by its own controller that is connected via an Ethernet network to communicate via remote procedure calls. For processing the data generated by sensors as well as process parameters (e.g., motor speed), the high throughput distributed messaging system Apache Kafka is used and Apache Cassandra is installed as database.

The overall manufacturing process is currently designed as a cycle to simulate a mass production environment. The process starts from the high-bay warehouse where workpieces are dispensed and transported to the multi-processing station – the oven and the milling machine. After processing, they are sorted by color, transported by the vacuum gripper robot and finally stored in the high-bay warehouse where the process starts again. As can be seen from the transport routes depicted as dotted lines in Fig. 2, the model also provides the option for executing manufacturing processes in a more flexible way as typical for Industry 4.0 mass customization [10,14].

3.2 Use Cases

In this subsection, we introduce two typical Industry 4.0 use cases in which we want to investigate the potential of an ontology. These are: Flexible Production Processes, Predictive Maintenance (PredM), and their interrelation towards the direction of the development of a CPS in which the resolution of errors in production processes is an important aspect [15].

Flexible Production Processes Essential aims of Industry 4.0 are to increase the flexibility to react on individual customer requirements and thus to produce customized products or to optimize efficiency in terms of the consumption of raw materials or energy from manufacturing processes. In order to achieve this, it is necessary to execute manufacturing processes in a more dynamic fashion [10]. However, this process requires knowledge about which machines can perform similar tasks and what their utilization rate is, which machines are reachable

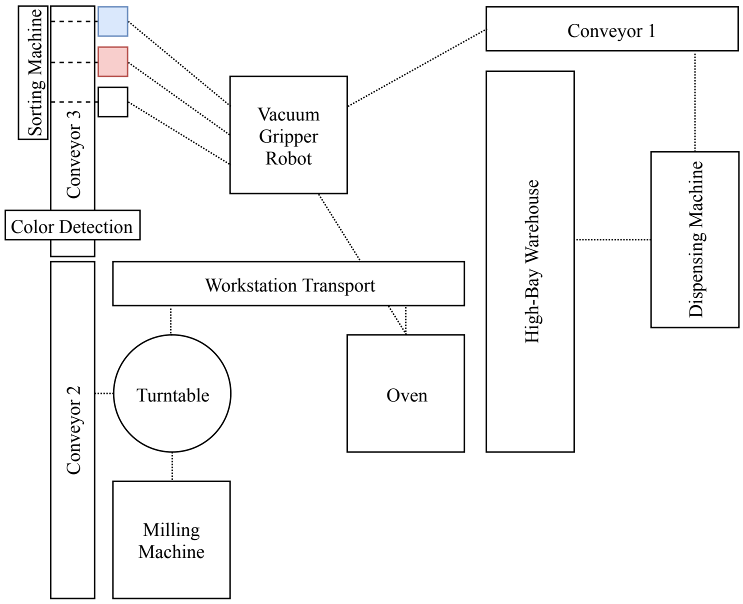

Figure 2. Schematic Illustration of the FT Factory Simulation Model

from the current position of the product and how the production processes can be carried out in an optimized way.

Condition Monitoring Analysis PredM [27] aims at foreseeing a breakdown of the system by detecting early signs of an upcoming failure to make maintenance work more proactive. Thus, it is possible to fix errors in production before they are happening and therefore prevent cost-extensive down times. An ontology can support this process by representing machines with their individual components and by linking connected sensors and measurements (temporal and spatial relationships). Failure data from individuals of the same (machine-) class can be used to apply transfer learning for a more robust prediction model. Furthermore, ontologies are used to solve the interoperability issue by relating information from heterogeneous sources to enhance condition monitoring data with contextual information such as from control software or other related systems [24].

Integration and Interrelation In the event of an inconsistency, e.g., caused by a bearing failure detected by unusual vibration patterns, the affected component or machine can be determined. Thus, the faulty part can be replaced in time – not too early, but also not too late. If an unexpected failure occurs in the production process or scheduled maintenance is performed, some machines or transport routes can be temporarily unavailable. In this case, currently running production processes or already planned processes may not be executed as scheduled. By using an ontology, similar machines or alternative transport routes can be identified to keep production processes running. Thus, an ontology provides the foundation for the use of planning techniques (e.g., for flexible process adaptations of unanticipated exceptions [20]) or generally for the integration of Internet of Things-based data such as sensor data streams with (business) process management (e.g., to support employees during work with mobile user guidance [26]).

4 FTOnto: Domain Ontology for a Fischertechnik Simulation Production Factory

The structure of this section follows the methodology presented in Sect. 4.1. In this section, the development process of the ontology and the underlying methodology are presented. Afterwards, the requirements for the ontology to be developed are specified and the developed ontology and its constituents are described in detail[2]. Finally, in Sect. 4.4 an evaluation is carried out to determine the suitability of the developed ontology.

4.1 Development Process

The development process of the ontology follows the well-known methodology of Sure et al. [28] for ontology development. Figure 3 depicts the ontology development process schematically. In the *kickoff* phase, the requirements for the

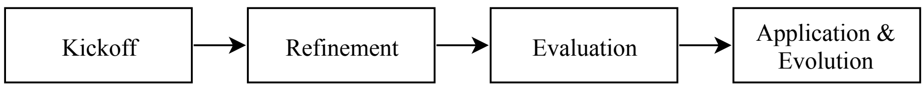

Figure 3. The Ontology Development Process by Sure et al. [28]

ontology to be developed have been identified (see Sect. 4.2). Furthermore, the concepts to be developed and the relationships between them have been determined and existing ontologies have been investigated for reuse. The second phase *refinement*, has been performed in a *top-down* fashion to expand and to elaborate the rough concepts and relations. In this process, the existing upper ontology *MASON* has been used for refinement. As a result, a prototype of the ontology has been created (see Sect. 4.3). In the *evaluation* phase, the ontology has been checked for conformity and consistency (see Sect. 4.4). For this purpose, it is demonstrated how an exemplary machine from the FT Factory is represented in the ontology and whether the specified requirements are satisfied. The last phase *Application & Evolution* aims at using the developed ontology in research and to further improve the ontology. Thus, future work is discussed in Sect. 5.

[2] The basic components of the ontology were developed and implemented in a student research project at Trier University by Christian Badouin and Marcel Mischo.

4.2 Requirements

In Sect. 3.2, we presented use cases in which the usage of an ontology is valuable. These use cases necessitate *requirements (RQs)* to be met by the ontology. In this section, requirements are derived from the use cases and are presented in the following:

RQ 1 – Machine Similarity: Similar machines and machine components as well as the similarity between their executable capabilities is required to be represented.

RQ 2 – Asset Availability: A relationship should exist between a failed machine and its impact on the manufacturing processes so that the affected resources could be identified.

RQ 3 – Transport Routes: The transport possibilities between machines for the handling of workpieces should be represented. It is necessary to ensure flexibility in the execution of manufacturing processes.

RQ 4 – Machine-Sensor Relationship: The relationships between machines and sensors need to be modeled so that signs of failures in a sensor signal can be related to the monitored machine.

4.3 Description of the Ontology

Since *CDM-Core* contains concepts such as people and geo-locations that can be useful in a real factory but are not needed in our case, we decided to build FTOnto from scratch with *MASON* and the *Sensor, Observation, Sample, and Actuator (SOSA)* ontology, which is a more compact version of SSN as the foundation. Hence, we are using a subset of CDM-Core, which is built on just two ontologies and consequently results in fewer classes and a more straightforward structure. FTOnto's top structure begins with the main classes *Manufacturing Concept* from *MASON*, the main classes of *SOSA* as well as parts of ontologies proposed by Cheng et al. [6] for flexible manufacturing based on web services. In the following, the refinements of FTOnto are explained.

Manufacturing Concept The class *Manufacturing Concept* contains the three head concepts of the *MASON* ontology: *Entity*, *Operation*, and *Resource*. To model the physical constituents of our FT Factory, most refinements are made as new subclasses of the class *Resource* namely *Machine resource* and are shown in the class hierarchy of Fig. 4. A subclass *Workstation* with a further subclass for each of the four workstations of the FT Factory is added. Since a machine consists of several actuators and sensors, the class *Machine component* with the subclass *Directly addressable* for modeling actuators and sensors as the lowest level and the class *Indirectly addressable* to model intermediate concepts, which are parts of the high-level concept *Workstation*.

The class *Entity* is extended by the class *Workpiece* for representing the workpieces used in the cycle of the model factory. Possible transport routes of workpieces on the conveyor belts, the turntable or with the vacuum gripper as shown in Fig. 2 are represented as instances of the class *Handling*.

Besides classes, we added several abstract roles (object properties) to relate instances with each other and concrete roles (data properties) to connect individuals with data values. For example, to express that an instance of the class *Conveyor belt* is driven by a motor instance, the abstract role *is actuated by* was introduced to relate both instances and the concrete role *has default speed* can be used to model that the motor is normally driven with a speed value of 512.

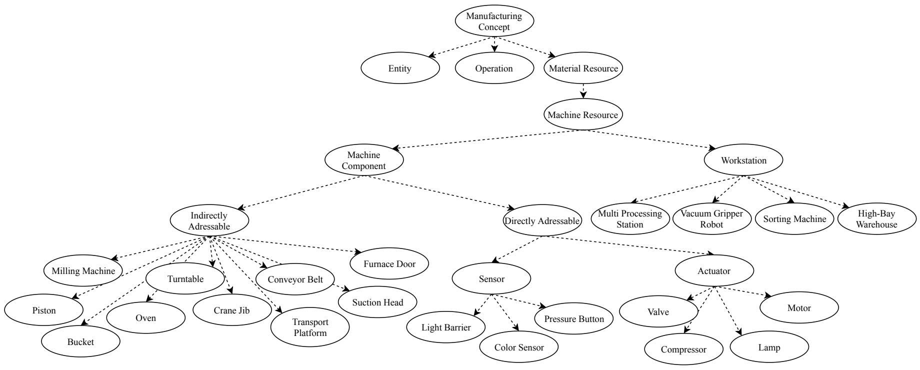

Figure 4. Part of the Class Hierarchy after Refinement of the Class *Machine resource* from *MASON* Ontology

SOSA Classes The SOSA ontology is required to describe the relationships between sensors and actuators as well as the measured data. We align the ontology of the manufacturing system with SOSA through the following class equivalence axioms:

$$\text{MASON.Sensor} \equiv \text{SOSA.Sensor}$$
$$\text{MASON.Actuator} \equiv \text{SOSA.Actuator}$$
$$\text{MASON.Indirectly Addressable} \equiv \text{SOSA.Platform}$$

For example, a light barrier is an instance of the class *Sensor* and mounted on some instance from the MASON class *Conveyor belt* to measure some *Feature of Interest*, e.g., the arrival of a workpiece, by obtaining instances of the class *Observations* where the class *Result* contains the measurement value.

Process and Service Ontology We use parts of the approach for web service integration for a flexible manufacturing system by Cheng et al. [6]. We remodeled their process ontology with the classes *Process* and *OperationSequence* that are related with the object property *hasArray* and linked it to *MASON* by changing the range of the object property *hasOperation* to the *MASON* class *Operation*.

To associate products with their manufacturing process, we used the object property *hasProcess* to relate *MASON*'s *Entity* class, which is designed to model products, to a process of the *Process* class. To provide a service oriented execution of manufacturing processes, we also remodeled their service ontology. Thus, each operation from *MASON* is connected by the object property *isRalizedBy* to a service that has a description and an URI.

4.4 Evaluation

This subsection presents an example of the semantic description for the model of a milling machine from the previously described Fischertechnik factory simulation (see Sect. 3.1). The simplified model of a milling machine is framed with green color in Fig. 5 and our semantic description is depicted as a graph in Fig. 6. By describing the graph, we would like to briefly address the requirements that have been specified in Sect. 4.2. In addition, the ontology was checked with *OntOlogy Pitfall Scanner! (OOPS!)*[3] for correctness.

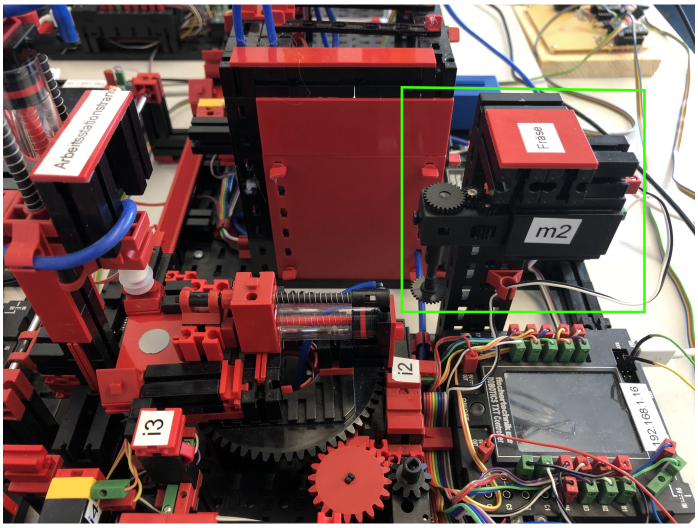

Figure 5. The Milling Machine in the FT Factory

Classes are surrounded by an orange circle and instances by a purple rectangle. The dashed arrow between both states that the instance is from the type of this class. For example, *MPS_MillingMachine* is the instance of the class *MillingMachine*. This implementation satisfies *RQ 1* that similar machines must be identifiable, which can be determined by the relationship between instances and their classes since instances of the same class can be considered to be similar.

[3] http://oops.linkeddata.es/

In addition, *MPS_MillingMachine* is driven by *MPS_Motor2*, which is modeled through the arrow that represents the property *actuates*. Moreover, the motor is controlled by *MPS_Machine_Controller* as indicated by the arrow labeled with *controls*. All three previously mentioned instances are part of the *MPS_MultiProcessingStation*, which is modeled by the property *has component*. Moreover, the *MPS_MillingMachine* instance provides a *Milling_Service* that enables a *Milling_Operation*. The presented relations are useful with regard to *RQ 2* so that in the case of a failure, relationships assist to determine which parts of the factory and corresponding services are affected. For example, if the motor of the milling machine fails, the milling machine is not working properly and consequently not its provided *Milling_Service*. A service enables an operation that is part of an operation sequence that in turn is part of a process instance that represents the manufacturing process of a product [6]. Each operation has a relationship to the subsequent operation (e.g., *Transport_From_Milling*). By adding a start and end position to each instance of a transport route, it is possible to find alternative routes and thus to enhance flexibility (see *RQ 3*).

Additionally, the motor *MPS_Motor2* is related via the property *hosts* with a sensor *AccSensor_ADXL345* that observes its vibration for condition monitoring purposes. This SOSA property allows to model the relationship between the milling machine and the sensor that monitors its condition (see *RQ 4*).

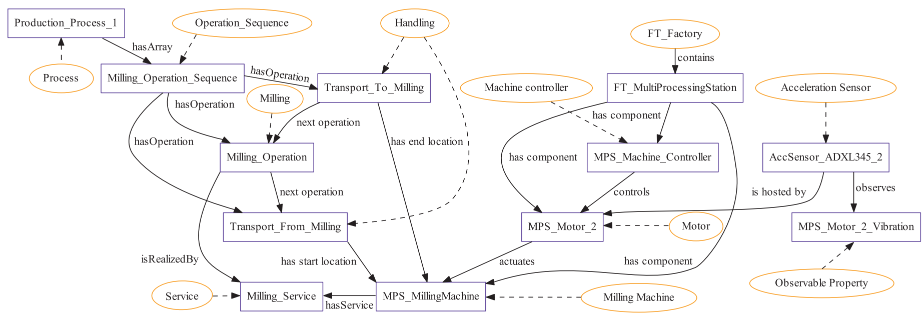

Figure 6. Part of FTOnto with Focus on the Semantic Annotations of the Milling Machine as Graph

5 Conclusion and Future Work

This paper investigates the development of an ontology to represent a Fischertechnik manufacturing simulation model by reusing existing ontologies. Simulation factories are a common method used for research purposes to investigate developed artifacts under laboratory conditions to examine their suitability and before they are potentially used in practice. The developed ontology contributes to support future research with simulation environments. It is intended to provide the developed ontology available for download under https://iot.uni-trier.de.

In future work, we investigate how the developed ontology can be further improved for our research purposes, e.g., by adding a context module to capture situation changes adequately such as process states [3,25]. This is especially important when controlling the execution of manufacturing processes. Furthermore, we plan to expand the FT Factory model with additional and redundant machines to provide similar services. Thus, it is possible to facilitate the use of process adaptation techniques in case of a failure. In addition, we plan to implement semantic web services to encapsulate the functions of workstations and thus to execute arbitrary manufacturing processes in our simulation environment. In this context, event-based ontology updates should also be examined (e.g., [6,23]). Finally, the combination of semantics with machine learning for PredM is investigated.

Acknowledgments. This work is funded by the German Research Foundation (DFG) under grant No. BE 1373/3-3.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications **7**(1), 39–59 (1994)
2. Ameri, F., Dutta, D.: An Upper Ontology for Manufacturing Service Description. In: Volume 3: 26th Computers and Information in Engineering Conf. ASME (2006)
3. Cao, Q., Giustozzi, F., Zanni-Merk, C., de Bertrand de Beuvron, F., Reich, C.: Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach. Cybern. Syst. **50**(2), 82–96 (2019)
4. Cao, Q., Zanni-Merk, C., Reich, C.: Ontologies for Manufacturing Process Modeling: A Survey. In: Int. Conf. on Sustainable Design and Manuf. pp. 61–70. Springer (2018)
5. Chandrasekaran, B., Josephson, J., Benjamins, V.: What are ontologies, and why do we need them? IEEE Intell. Syst. **14**(1), 20–26 (1999)
6. Cheng, H., Xue, L., Wang, P., Zeng, P., Yu, H.: Ontology-based web service integration for flexible manufacturing systems. In: 15th IEEE Int. Conf. on Industrial Informatics, INDIN, Germany. pp. 351–356. IEEE (2017)
7. Compton, M., et al.: The SSN ontology of the W3C semantic sensor network incubator group. J. Web Semant. **17**, 25–32 (2012)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 Web Ontology Language Primer. W3C Recommendation (December 2012), available at https://www.w3.org/TR/2012/REC-owl2-primer-20121211/
9. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman and Hall/CRC Press (2010)
10. Kagermann, H., Helbig, J., Wahlster, W.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; Final report of the Industrie 4.0 Working Group. Forschungsunion (2013)
11. Klein, P., Bergmann, R.: Generation of Complex Data for AI-Based Predictive Maintenance Research With a Physical Factory Model. In: 16th Int. Conf. on Informatics in Control Automation and Robotics, ICINCO, Czech Republic. pp. 40–50. SciTePress (2019)

12. Klinker, G., et al.: FixIt: An Approach towards Assisting Workers in Diagnosing Machine Malfunctions. In: Proc. of the IUI-CADUI*04 Workshop on Exploring the Design and Engineering of Mixed Reality Syst., Portugal. CEUR-WS.org (2004)
13. Lang, S., Reggelin, T., Jobran, M., Hofmann, W.: Towards a Modular, Decentralized and Digital Industry 4.0 Learning Factory. In: Sixth Int. Conf. on Enterprise Syst., Cyprus. pp. 123–128. IEEE (2018)
14. Lasi, H., Fettke, P., Kemper, H., Feld, T., Hoffmann, M.: Industry 4.0. BISE **6**(4), 239–242 (2014)
15. Lee, J., Bagheri, B., Kao, H.A.: A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. MFGLET **3**, 18–23 (2015)
16. Lee, J., Kao, H.A., Yang, S.: Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. Procedia CIRP **16**, 3–8 (2014)
17. Lemaignan, S., Siadat, A., Dantan, J.Y., Semenenko, A.: MASON: A Proposal For An Ontology Of Manufacturing Domain. In: IEEE Workshop on Distributed Intell. Syst.: Collective Intelligence and Its Applications. pp. 195–200. IEEE (2006)
18. Lin, H.K., Harding, J.A.: A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. Computers in Industry **58**(5), 428–437 (2007)
19. Liu, B., Wu, J., Yao, L., Ding, Z.: Ontology-based Fault Diagnosis: A Decade in Review. In: Proc. of the 11th Int. Conf. on Computer Modeling and Simulation. pp. 112–116. ICCMS, ACM, USA (2019)
20. Marrella, A., Mecella, M.: Cognitive Business Process Management for Adaptive Cyber-Physical Processes. In: BPM 2017 Int. Workshops, Spain, Revised Papers. pp. 429–439. LNBIP, Springer (2018)
21. Mazzola, L., Kapahnke, P., Vujic, M., Klusch, M.: CDM-Core: A Manufacturing Domain Ontology in OWL2 for Production and Maintenance. In: Proc. of the 8th Int. Joint Conf. on Know. Discovery, Know. Engineering and Know. Management - Vol. 2: KEOD, Portugal. pp. 136–143 (2016)
22. Pajic, M., Chernoguzov, A., Mangharam, R.: Robust architectures for embedded wireless network control and actuation. ACM TECS **11**(4), 82:1–82:24 (2012)
23. Puttonen, J., Lobov, A., Lastra, J.L.M.: Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services. IEEE TII **9**(4), 2349–2359 (2013)
24. Schmidt, B., Wang, L., Galar, D.: Semantic framework for predictive maintenance in a cloud environment. Procedia CIRP **62**, 583 – 588 (2017)
25. Scholze, S., Barata, J.: Context Awareness for Flexible Manufacturing Systems Using Cyber Physical Approaches. In: Technological Innovation for Cyber-Physical Syst. pp. 107–115. Springer (2016)
26. Schönig, S., Jablonski, S., Ermer, A.: IoT-basiertes Prozessmanagement: Mobile Benutzerführung in der digitalen Fabrik. Informatik Spektrum **42**(2), 130–137 (2019)
27. Selcuk, S.: Predictive maintenance, its implementation and latest trends. Proc. Inst. Mech. Eng. B **231**(9), 1670–1679 (2017)
28. Sure, Y., Staab, S., Studer, R.: Ontology Engineering Methodology. In: Handbook on Ontologies. pp. 135–152. Int. Handbooks on Inf. Syst., Springer (2009)
29. W3C OWL Working Group: OWL 2 Web Ontology Language Document Overview (Second Edition) - W3C Recommendation 11 December 2012
30. Wang, Y.M., Yin, H.L., Xiao, N.F., Jiang, Y.R.: Internet-based remote manipulation and monitoring of an industry robot in advanced manufacturing systems. Int. J. Adv. Manuf. Technol. **43**(9-10), 907–913 (2008)