Toward a Knowledge-Based Data Backbone for Seamless Digital Engineering in Smart Factories

Alexander Perzylo, Ingmar Kessler, Stefan Profanter, Markus Rickert

***Abstract*—Digital transformation efforts in manufacturing companies bear the potential to reduce product costs and increase the flexibility of production systems. The semantic integration of data and information along the value chain enables the automated interpretation of interrelations between its different aspects such as product design, production process and manufacturing resources. These interrelations can be used to automatically generate semantic process descriptions and execute corresponding robot motions. An initial one-time effort to model the required knowledge of a particular application domain can make the manufacturing of high-variant products in small batches or even lot size one production more efficient.**

This paper introduces a knowledge-based digital engineering concept to automate engineering and production activities without human involvement. The concept was integrated and evaluated in a physical robot workcell where automotive fuse boxes are autonomously fitted with different fuse configurations.

I. Introduction

Digital transformation is currently one of the key challenges in the manufacturing industry, when it comes to streamlining and automating internal processes. The core goals include establishing data access to technical systems, sharing data, and increasing the effectiveness and efficiency of data processing. Most companies already use digital data such as PDF files, spreadsheets, and other documents that often share different types of identifiers. But despite being digital data representations, they do not provide the semantic meaning of the data or an integrated view on multiple data sources. Thus, the potential of a proper digital transformation strategy is not realized.

Manufacturing companies in particular face many challenges. They have to optimize product costs and maximize the flexibility of their production systems to stay competitive while addressing individual customer requests. This leads to high-variant products and small-batch production. However, the manual reconfiguration and operation of production systems are major cost drivers. Automating these activities is therefore essential when competing in a global market [1].

The production phase is a key aspect for assessing the product costs and performance criteria of a manufacturing value chain, but it is only one of several interrelated phases. Product design and development take place before the start of production. Production processes have to be specified and checked for compliance with requirements derived from product features. Production systems are engineered to meet the requirements of both the production process and the product. Process data generated during individual production runs have to be interpreted with respect to these requirements and suitable performance criteria have to be derived and evaluated. Manufacturing companies typically have dedicated departments for these activities, but no semantic representation of data and information that is shared among them and the production system. Product lifecycle management (PLM) tools provide meta-level descriptions of various data and file formats, but lack a deep semantic understanding of how they are interconnected and the implications thereof.

A. Perzylo, I. Kessler, S. Profanter and M. Rickert are with fortiss, An-Institut Technische Universität München, Munich, Germany.

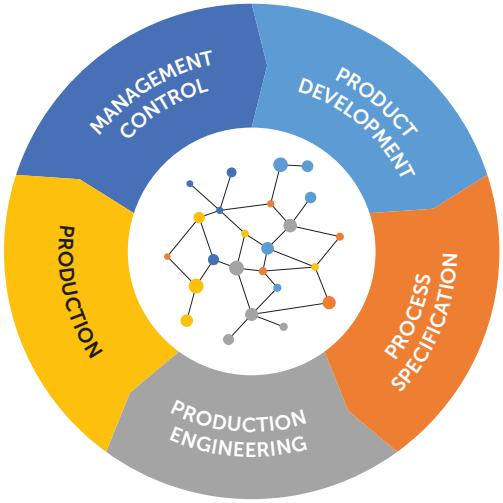

Fig. 1: The knowledge-based Data Backbone in the center represents and links data from different engineering activities along the value chain of manufacturing companies.

We propose a knowledge-based digital engineering concept for manufacturing companies that involves semantically describing and integrating all relevant knowledge of a product, its production process, and related manufacturing resources. Ontologies are used to model and combine commonsense, domain, and application knowledge to provide a deeper understanding of the task at hand. Based on this context-rich semantic description, production-related engineering activities can be automated.

In this paper, we present our knowledge-based Data Backbone concept (Fig. 1) that semantically integrates the introduced activities along the value chain. Furthermore, we introduce a realistic use case that has been automated in a physical robot workcell (Fig. 9). There, we show how even lot size one production can be automated, starting from an order and ending with the execution of generated robot motions, by accessing and interpreting relevant semantic information.

II. Related Work

The work in this paper is based on two core principles, i.e., the explicit representation of knowledge using ontologies and the integration of manufacturing resources using skill models.

Skills can be seen as a tool-centric approach to process modeling and execution that simplifies the abstraction of functionalities provided by hardware and software components [2], [3]. OPC UA (Open Platform Communications Unified Architecture) is a protocol for device integration and information exchange in modular production systems developed by the OPC Foundation. In [4], a standardized OPC UA skill model is proposed to integrate and control industrial robots and tools in a hardware- and manufacturer-independent manner. This skill model provides, e.g., generic move skill types, which can be hierarchically composed to higher-level functionalities. Using automatic device discovery, as described in [5], these device components can be dynamically integrated on the shop floor.

In recent years, there has been active research in knowledge representation for industrial automation and robotics. Many approaches follow the product-process-resources paradigm (PPR) [6], in which products [7] and product designs [8], production processes [9], and manufacturing resources [10] are modeled. The combination of skills and explicit knowledge representation is often used to reduce the complexity of programming production systems [9], [11], [12]. In [13], a CAD-based instruction of assembly tasks is described. The authors suggest a system architecture, where geometric constraints are specified on an application layer to describe assembly steps. They are mapped to the skills of a workcell and then executed and monitored using restricted finite state machines.

Rosen et al. emphasize the increasing importance of autonomous manufacturing and meaningful digital representations of products and manufacturing resources [14]. A prerequisite for increasing the autonomy in manufacturing is access to relevant data and information and a formal representation that encodes their semantic meaning. This includes semantic access to data [15], [16] and the extension of PLM systems [17]. In [18], the authors present a combination of semantic process knowledge and OPC UA-enabled devices to create a self-organizing production system. Given the semantic context information from PPR models and general automation knowledge, autonomous production can be extended with the automated annotation of process data and derivation of key performance indicators (KPIs) [19], [20], [21].

In contrast to the mentioned papers and our own previous work, the approach presented in this paper aims at replacing the manual instruction of production systems with a knowledge-driven synthesis of process descriptions. We propose a product-centric process modeling paradigm that differs from tool-centric approaches, which are based solely on the sequencing of specific skill invocations. Product-centric process models contain hardware-independent descriptions of the required manufacturing steps leading to the creation of a given product. During the deployment of such a device-agnostic process description, suitable hardware and software components are identified based on a formal description of their capabilities [22]. As a result, the same process model can be reused in different production environments.

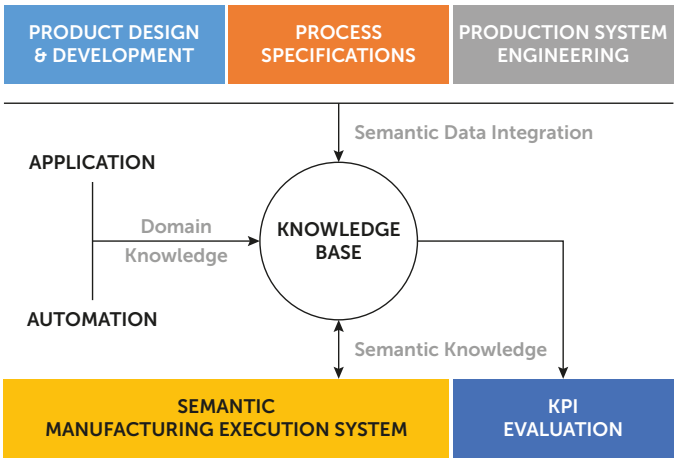

Fig. 2: Semantic representations of various types of knowledge are integrated in a shared knowledge base to provide semantic context knowledge for our semantic MES in order to autonomously perform and evaluate manufacturing tasks.

III. Explicit Knowledge Representation

Industrial automation solutions are characterized by a large engineering effort and tailored to particular hardware and software components for specific use cases. Furthermore, a lot of commonsense and domain knowledge is often encoded in the software implementations that drive the systems. Consequently, the knowledge is only implicitly represented and hidden in source code. Sharing the knowledge is complicated by the use of various programming languages and the interweaving of task logic and control code. In our concept, knowledge is separated from software implementations and described in ontologies using a formal knowledge representation language to make it reusable and shareable. The explicitly represented knowledge can be more easily maintained and flexibly extended. As a result, software implementations can be designed as generic components that are configured for specific hardware and use cases via semantic models.

Our knowledge-based digital engineering concept uses the OWL 2 Web Ontology Language [23] as its formal knowledge representation language. OWL was developed by the World Wide Web Consortium (W3C) for the Semantic Web. Due to its wide applicability and many available software tools, it can be used in other areas such as the domains of industrial automation and robotics as well. OWL provides methods to formally describe entities as classes, instances of these classes called individuals, and properties that define the relations between them or connect them with literal values such as numbers or strings. New complex classes and relations can be created by combining existing entities. Globally unique names are assigned to new complex entities to hierarchically

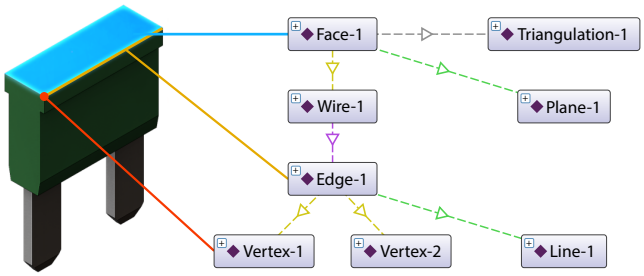

Fig. 3: Simplified excerpt from a deep semantic model of a blade fuse's geometry based on the OntoBREP ontology.

extend the existing vocabulary of an ontology. Due to OWL's logical formalism, explicitly modeled knowledge can not only be queried, but also interpreted by semantic reasoning software to logically infer additional implicit knowledge.

IV. Knowledge-driven Digital Engineering

Our knowledge-driven engineering approach aims at semantically describing all relevant data and information that is required to automatically generate suitable sequences of robot, tool, and sensor activities for achieving a given production goal. One key aspect is the use of a common representation for describing diverse data from heterogeneous sources and different types of knowledge (Fig. 2). These descriptions do not merely provide meta-information, but instead fully encode the actual content and its meaning. This includes knowledge of the product to be built, the production process that creates the product, the manufacturing resources that perform the process, and general knowledge of the automation and application domains. The semantic integration of certain kinds of information can be automated, while for others it still has to be carried out manually. As the manual effort only needs to be expended once and the resulting models can be shared and reused, the initial effort can lead to efficiency gains in the long run.

We use OWL to model all relevant entities in ontologies. In OWL every entity has an Internationalized Resource Identifier (IRI), which can be used to unambiguously reference the entity and to link it with other entities. This concept of linked data in combination with *deep* semantic models makes it possible to describe relationships across different knowledge domains and levels of granularity.

A. Semantic Data Integration

Regular CAD models, i.e., STEP or IGES files, of products, parts, manufacturing resources, and workcells can be automatically converted into our ontology-based boundary representation (OntoBREP) using a self-developed software component [7]. In this geometry representation, every vertex, edge, face, and solid is represented in OWL as an individual. They can be annotated with additional information and linked with, e.g., the description of a related production process.

Fig. 3 visualizes an excerpt of a geometry representation based on the OntoBREP ontology. The top face (Face-1) of the depicted automotive blade fuse's geometry is defined by an infinite plane (Plane-1). The individual Wire-1 is a topological BREP entity that contains a total of four edges that make Face-1 a finite rectangular-shaped surface. The highlighted edge (Edge-1) is defined by a line (Line-1) bounded by two vertices (Vertex-1 and Vertex-2). In addition to this boundary representation, which contains exact mathematical models of geometries, the individual Triangulation-1 provides a triangulation of Face-1, which can be used for visualization purposes.

Another example for automated semantic integration is the transformation of industry-standard information models of manufacturing resources to OWL representations. In our concept, OPC UA is used as the communication technology between hardware and software components in the overall production system. In OPC UA, components describe their functionalities and internal states with so-called node sets, which can be remotely browsed by other components. OPC UA node sets are defined based on XML schemas and can be automatically transformed to OWL representations [10].

Since most companies do not have formal descriptions of their production processes, creating semantic process descriptions is still mostly a manual activity. Typically, plain texts and spreadsheets need to be analyzed and converted into a sequence of production steps with adequate parameters. For each type of production step and parameter, OWL classes and properties are created or reused. The resulting vocabulary is applied in the modeling of a workcell-independent description of a production process. This abstract process can be instantiated for specific workcells and individual production runs in an automated manner.

The semantic process description can be seen as a hardware-agnostic formalization of the required production steps to create a given product. It contains a high-level sequence of abstract tasks, including their types and parameters, that are described with respect to abstract objects, including their types and properties. During the deployment of an abstract process to a manufacturing workcell, individual tasks are matched to compatible skills that are provided by the workcell's hardware and software components. Objects and other parameters are similarly mapped to suitable objects and other entities in

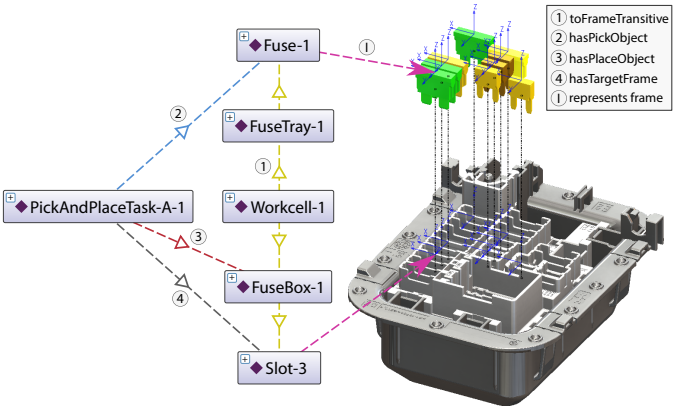

Fig. 4: Task parameters are represented semantically as objects and frames that are linked with other entities in ontologies.

the workcell. As a result, the same abstract process can be reused and may result in different skill invocations and specific parameters for different workcells (Fig. 4).

B. Knowledge Base

The Knowledge Base (KB) component is responsible for the central storage and interpretation of and access to the semantic knowledge in the system. For this purpose, it uses the triple store and OWL 2 RL [24] inference capabilities of Ontotext GraphDB[1]. The KB uses the open-source OPC UA stack Eclipse Milo[2], so that other hardware and software components in a workcell can directly communicate with it via OPC UA. For each use case, such as a robot workcell for mounting fuses, both general OWL ontologies as well as specialized ones, e.g., about application domain, production process and workcell, are loaded into the KB.

In order to make the system more flexible and to separate knowledge from source code, the ontologies may contain SPARQL 1.1 (SPARQL Protocol and RDF Query Language) requests, i.e., SELECT and CONSTRUCT queries or DELETE and INSERT updates. These requests are executed dynamically to query and manipulate knowledge in the KB during the procedures described in Sections V-B, V-C and V-D.

Each modeled SPARQL request is defined by its IRI, type, parameters, and request string. This representation is based on the textual SPARQL forms in SPIN (SPARQL Inferencing Notation) [25] and SHACL (Shapes Constraint Language) [26]. The KB mainly provides three OPC UA methods for executing SPARQL requests and returning their results as JSON [27] for SELECT queries and as N-Triples [28] for CONSTRUCT queries.

- *sparql-request(repoId, requestString)* executes the given SPARQL request string in the given repository.
- *sparql-command(repoId, requestIri, parameters)* executes the SPARQL request identified by the given IRI with the given parameters.
- *linked-sparql-command(repoId, resourceIri, propertyIri, parameters)* executes all SPARQL requests that are linked with the given resource via the given property.

An OWL class (using owl:hasValue restrictions) or individual can be linked with multiple requests via the same property. Thus, when a request is defined for each type in a class hierarchy, a set of requests can be inferred for an individual based on its type. On account of this, *linked-sparql-command* executes all linked requests, e.g., multiple updates, in an arbitrary but consistent order. For multiple CONSTRUCT queries the union of their results is returned.

C. Semantic Manufacturing Execution System

The semantic Manufacturing Execution System (sMES) controls the other hardware and software components in a particular workcell and is managed by a superordinate shop floor MES. The shop floor MES is not described in further detail, since it may be any proprietary implementation or industrial solution. Its role is to gather information on the next production run and trigger the sMES with the corresponding product ID.

[1]https://www.ontotext.com/products/graphdb/

[2]https://github.com/eclipse/milo

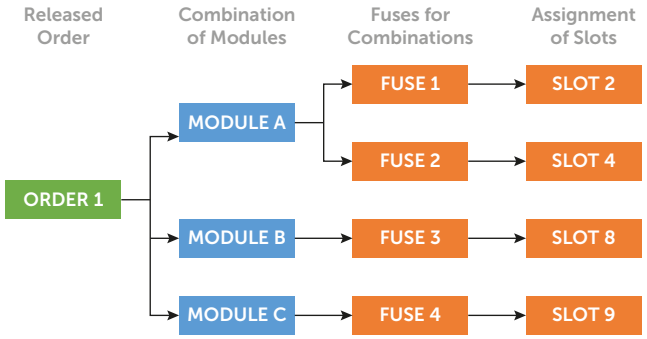

Fig. 5: Steps leading up to the creation of a process description for equipping an automotive fuse box – from a released order to the corresponding fuse configuration and slot assignment.

The sMES component is responsible for carrying out a production process. This includes mapping abstract processes to specific workcells (Section V-B) and executing the resulting specific processes (Section V-C). During this deployment, the semantic process description in the KB is queried via OPC UA methods (Section IV-B) in order to set the skills' parameters prior to their invocation.

All device components in the system implement a generic skill model [4] based on OPC UA's ProgramStateMachineType with predefined states and transitions. The execution of a skill is monitored based on the associated state machine. In addition to the generic skill interface, all device components have to implement OPC UA Local Discovery Services [5]. As soon as a newly plugged-in device is detected, the sMES triggers its internal skill detector to browse the remote server for provided skill types. Thus, at all times an up-to-date list of available skills is maintained, which can be used in the mapping procedure.

As described in [4], lower-level skills can be grouped hierarchically to compose new and more complex skill types. For example, a Pick-and-Place skill type is composed of robot movement and gripper manipulation skills. This hierarchical composition leads to further abstraction of hardware functionalities and enables the intuitive modeling of complex skills.

V. From Order to Autonomous Assembly

This section shows how the introduced concepts can be applied to the knowledge-driven automation of equipping automotive fuse boxes with different fuse configurations.

A. Semantic Process Description

The steps leading up to the required information for modeling the semantic process descriptions are depicted in Fig. 5. Based on an order consisting of a list of requested modules, a module combination-specific wiring configuration is derived. Afterwards, suitable fuse types are selected for the modules.

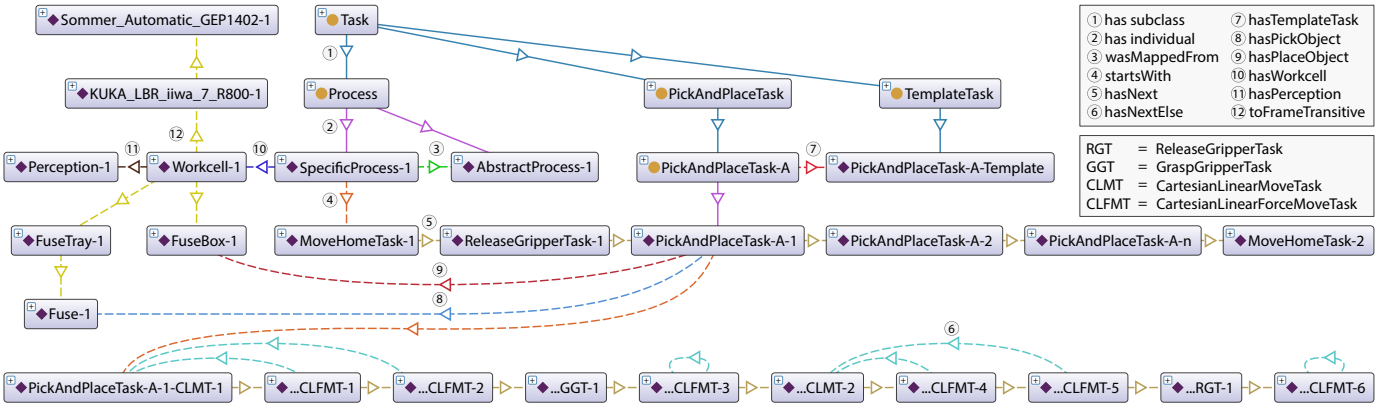

Fig. 6: Simplified overview of the OWL ontologies in the KB for a particular fuse mounting process and workcell. It shows mapped process SpecificProcess-1 and its subtasks MoveHomeTask-1, ReleaseGripperTask-1, PickAndPlaceTask-A-1, etc. It further visualizes a sequence of robot and gripper subtasks generated for PickAndPlaceTask-A-1 based on a modeled template.

In a last step, a mapping of fuses to slots in the fuse box is generated.

The semantic process description from Section IV-A contains not only knowledge about the process itself, but also links with other semantic descriptions that are relevant to the process's execution. This includes, e.g., the parts of a product, reusable production steps, or, once the process is deployed to a workcell, a robot and tool. The KB stores them in various contexts corresponding to their respective OWL ontologies. Part ontologies may contain, e.g., the ID, type, size, current rating, and color of fuses. Workcell ontologies may include hardware and software components as well as information about them such as a robot's workcell-specific home configuration. A separate perception context, which may be updated by object detection or inventory management systems, contains parts that dynamically enter and leave a workcell, e.g., fuses and fuse boxes.

```
PREFIX core: <http://www.fortiss.org/ont/robotics/core#>
SELECT ?specificPickObj ?abstractPickObj
WHERE {
 $specificProcess core:context ?context .
 $abstractTask core:hasPickObject ?abstractPickObj .
 { ?specificPickObj core:wasMappedFrom ?abstractPickObj .
   BIND(1 AS ?priority) }
 UNION
 { ?specificPickObj a core:Object , core:Specific .
   FILTER NOT EXISTS {
    ?abstractPickObj a ?pickObjType .
    FILTER (core:Abstract != ?pickObjType)
    FILTER NOT EXISTS {?specificPickObj a ?pickObjType} }
   BIND(2 AS ?priority) }
 FILTER EXISTS {GRAPH ?context {?specificPickObj a []}} }
ORDER BY ?priority ?specificPickObj
LIMIT 1
```

Listing 1: Simplified subquery from a SPARQL mapping update for PickAndPlaceTasks. It maps a pick object from an abstract process to an object in a specific workcell based on its types. Each pick object is an abstract parameter that is mapped only once and shared among tasks in the same process.

B. Mapping of Abstract Process to Specific Workcell

Since abstract processes are workcell-independent, they have to be mapped to a specific workcell first, in order to generate an executable specific process (Fig. 6). This mapping procedure is modeled using SPARQL updates that are called via OPC UA (Section IV-B) by the sMES at the beginning of the execution. First, initialization updates set up the specific process based on the given workcell, which mainly consists of creating a new context, setting metadata, and taking a snapshot of the workcell's perception context.

Secondly, beginning with the process itself, abstract tasks and parameters are mapped recursively to generate specific tasks and parameters using mapping updates (Listing 1) that are defined along the class hierarchy of the tasks. If a task can be matched directly to a skill of a device component in the workcell, the mapping procedure continues with the next task. Otherwise, if the task starts with a subtask or if the task has a TemplateTask for generating one, the subtask is mapped recursively first. TemplateTasks contain lower-level subtasks to achieve the intended overall effect given shared parameters such as pick object, place object, and target frame (Fig. 4).

C. Execution of Specific Process

When the sMES (Section IV-C) is called to execute a process, it first performs the process-to-workcell mapping (Section V-B). At that time, it also creates the process's tasks and their KPIs in its OPC UA address space (Section V-D).

The actual execution follows directly afterwards and begins recursively with the specific process itself. If a task was matched to an available skill, the sMES calls the KB to execute its parameter queries. Their results contain the parameter values for the invocation of the skill. Due to the generic skill interface, they can be parsed, processed, and mapped to corresponding OPC UA variables that are intended to hold the skill's parameters. The skill execution is then started via the state transition methods of its state machine. If a task was not matched to any skill, but it starts with a subtask, then the

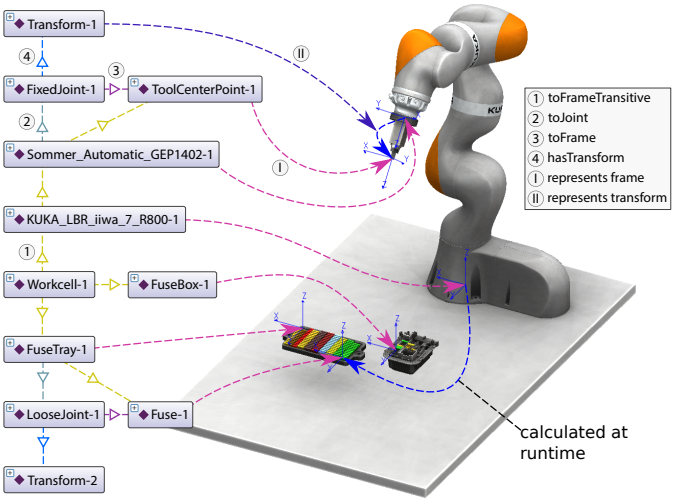

Fig. 7: Product parts, manufacturing resources and their layout in a workcell are represented semantically in ontologies.

subtask is executed recursively instead, e.g., as shown by Fig. 6 for the specific process itself and PickAndPlaceTask-A-1.

The sMES monitors the execution and calls the effect updates of successful tasks in the KB. The effect update of, e.g., a PickAndPlaceTask includes removing the pick object from its old (relative) location in the KB's semantic scene graph and inserting it at its new (relative) location (Fig. 7). For both successful and unsuccessful tasks the sMES calls their status updates to insert the monitoring results into the KB. After the end of a task, the sMES asks the KB which task should be executed next. If the task was successful, the next task according to the semantic process description is executed. Otherwise, the semantic process description may include error handling strategies. For instance, in a PickAndPlaceTask-A the execution can jump back to an earlier subtask in case of an error (Fig. 6) and the subtasks are rerun for a specified number of attempts.

D. Semantic Process Status

The semantic process status is part of the semantic process description in the KB and updated during the execution by the SPARQL status updates in Section V-C. A general status update for all task types sets the new status, e.g., OK/NOK, and internally calculates additional KPI-related information such as start time, end time, number of executions, number of errors, and the process's currently executed task. Some skill types have additional status updates for skill-specific information, e.g., the maximum measured forces during the execution of a CartesianLinearForceMoveSkill.

After calling a task's status updates, the sMES calls its status queries, since the status and KPIs may change due to calculations and completed subtasks. The status queries defined along the class hierarchy of the tasks include general status queries, e.g., for calculating and returning a task's duration, and specialized status queries, e.g., for returning the maximum measured forces. The KPIs and status queries of

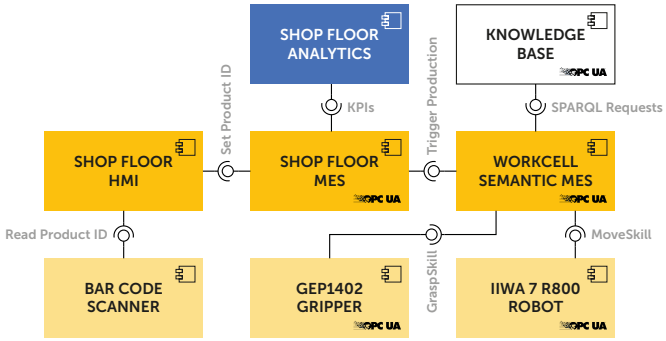

Fig. 8: Hardware and software components in the robot workcell provide OPC UA skills and services.

some task types are affected by their subtasks. The status of the PickAndPlaceTask for mounting a fuse includes the maximum measured forces during the place motion and the number of pick and place attempts that were necessary. Similarly, the status of the process itself automatically shows the maximum measured forces among any of its PickAndPlaceTasks.

The sMES parses, processes, and maps the status query results to OPC UA variables in its address space. These variables correspond to the KPIs of the tasks and enable OPC UA clients to browse the current semantic process status.

VI. Evaluation

The components and semantic descriptions of the system are largely independent of any specific use case or workcell to the extent that, e.g., the source code of the sMES does not know what a PickAndPlaceTask is and the source code of the KB does not even know what a task is. Therefore, software components such as the KB and consequently the sMES are configured with general and domain ontologies as well as abstract process and specific workcell ontologies. Device components provide skills for robots, tools, or other hardware by wrapping them in OPC UA servers [4].

In this way, a physical robot workcell (Fig. 9) was set up for the evaluation of the system using the architecture

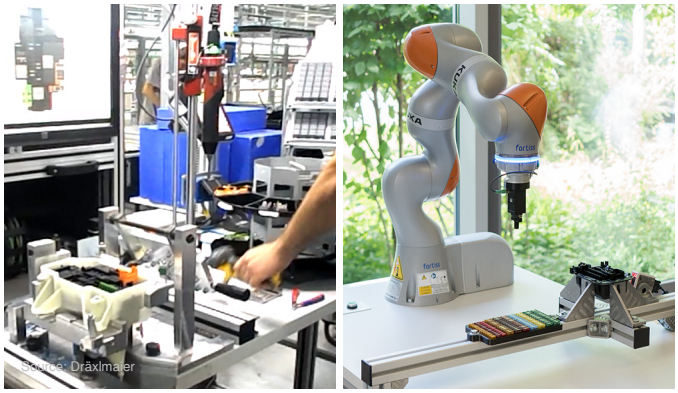

Fig. 9: Manual assembly station (left) and robot workcell (right) consisting of a robot, a parallel gripper, a barcode scanner, a tray with automotive fuses, and a fuse box fixture.

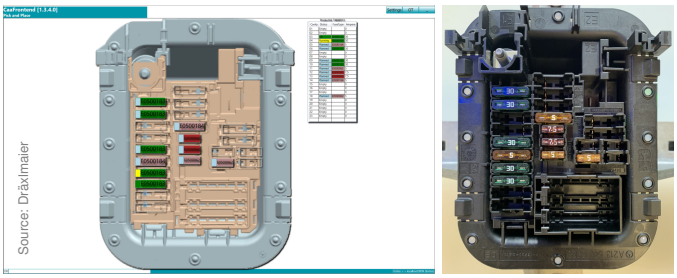

Fig. 10: Active slots in the fuse box shown in a HMI during production (left). OPC UA variables in the address space of the sMES provide the status based on the semantic process description in the KB. Fuse box fitted by process P2 (right).

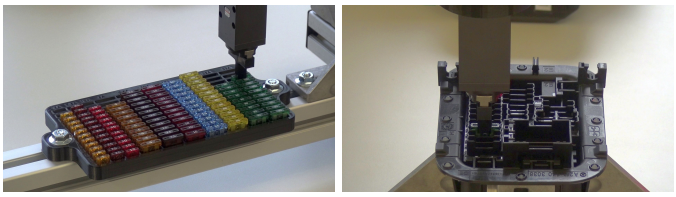

Fig. 11: Robot mounts each fuse by picking it from a tray (left) and placing it in the assigned slot of a fuse box (right).

and components shown in Fig. 8. The workcell contains a robot (KUKA LBR iiwa 7 R800), on which an electric parallel gripper (Sommer Automatic GEP1402) was mounted. The gripper fingers were designed after the shape of manual pliers for automotive blade fuses and created using additive manufacturing. A tray for supplying different types of blade fuses and a fixture for fuse boxes were installed. A bar code scanner of an automotive supplier reads the product ID on each empty fuse box that is placed in the fixture. An automotive supplier's human-machine interface (HMI) shows the fuse box's current status (Fig. 10). The shop floor MES of an automotive supplier is responsible for managing the production of different items and sends the product ID via OPC UA to the sMES. This triggers the fitting of a fuse box via the execution of a corresponding abstract process from the KB.

In the evaluation, the sMES was ordered to execute five different processes that contain fuse configurations provided by an automotive supplier. Table I lists for each process the number of fuses, the manual duration by a human worker, and the automated duration by the system from one execution. The human worker is only somewhat faster, although at this stage the goal was not to optimize the speed of the system, but to

TABLE I: Comparison of cycle times between the manual and automated mounting of blade fuses as measured by an automotive supplier.

Process	P1	P2	P3	P4	P5
Number of fuses	9	11	8	18	11
Manual assembly (in s)	121	128	118	151	128
Autonomous assembly (in s)	131	139	151	231	150

Fig. 12: An automotive supplier's shop floor analytics dashboard showing the maximum measured fuse mounting forces in Z in the end effector frame for each PickAndPlaceTask in one execution of process P1 (top) and across different assemblies for slot 9 (bottom). For the mounting of mini fuses (11–13, 18) higher maximum Z forces were measured than for regular ATO fuses (3–5, 9, 14).

investigate the feasibility of our knowledge-driven automation approach. A video of the experiment can be found online[3].

For the mounting of fuses (Fig. 11), the system relies on the force-torque sensors of the robot. Forces occurring during the execution of tasks are measured and logged together with other KPIs in the semantic process description in the KB. They can then be analyzed, visualized as depicted in Fig. 12, and interpreted with respect to maximum allowed forces. If the maximum forces are exceeded during the execution of a task, this information is stored in the semantic process description and an error handling strategy is performed (Section V-C). This is in addition to the robot's configured maximum forces that trigger an emergency stop. As an extension to the current system, the robot could provide an additional move skill for mounting fuses with a target force in addition to a target pose. This way, the robot would stop when a target force is exceeded making the fuse mounting process more robust and accurate.

VII. Conclusion

This paper describes a knowledge-based digital engineering concept for the semantic integration of diverse data and information along the value chain of manufacturing companies. The resulting OWL ontologies are used to increase the level of

[3]https://youtu.be/PtPd3YvTTzw

autonomy in skill-based manufacturing of high-variant products in small batches. We further introduced the core aspects of our production system, which consists of a knowledge base for the storage and interpretation of the ontologies and a semantic MES that manages the knowledge-based manufacturing using OPC UA device components and their provided skills. The proposed concept was evaluated in a physical robot workcell for the automated fitting of fuse boxes with different fuse configurations. Although required knowledge had to be modeled initially, the specific sequences of production steps for the robot were generated without human involvement. Based on the semantic process descriptions, suitable KPIs were automatically prepared during the production process. In its entirety, a knowledge-based Data Backbone was introduced that provides the grounds for increasing the level of autonomy of manufacturing systems.

ACKNOWLEDGMENTS

We would like to thank Franz Gleixner, Manfred Seitz, and Christian Veicht from Dräxlmaier for providing the use case and supporting the evaluation of the presented concepts.

The research leading to these results has been funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) under grant agreement no. IUK-1711-0033 in the project Data Backbone with project support from the Zentrum Digitalisierung Bayern (ZD.B).

REFERENCES

[1] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann *et al.*, "SMErobotics: Smart robots for flexible manufacturing," *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 78–90, Mar. 2019.

[2] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh *et al.*, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, Feb. 2016.

[3] S. Malakuti, J. Bock, M. Weser, P. Venet, P. Zimmermann *et al.*, "Challenges in skill-based engineering of industrial automation systems," in *Proc. of the IEEE Intl. Conf. on Emerging Technologies and Factory Automation*, Torino, Italy, Sep. 2018, pp. 67–74.

[4] S. Profanter, A. Breitkreuz, M. Rickert, and A. Knoll, "A hardware-agnostic OPC UA skill model for robot manipulators and tools," in *Proc. of the IEEE Intl. Conf. on Emerging Technologies And Factory Automation*, Zaragoza, Spain, Sep. 2019, pp. 1061–1068.

[5] S. Profanter, K. Dorofeev, A. Zoitl, and A. Knoll, "OPC UA for plug & produce: Automatic device discovery using LDS-ME," in *Proc. of the IEEE Intl. Conf. on Emerging Technologies And Factory Automation*, Limassol, Cyprus, Sep. 2017.

[6] A. Björkelund, H. Bruyninckx, J. Malec, K. Nilsson, and P. Nugues, "Knowledge for intelligent industrial robots," in *AAAI Technical Report SS-12-02, Designing Intelligent Robots: Reintegrating AI*, G. Konidaris, Ed., vol. SS-12-02. AAAI, 2012.

[7] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Hamburg, Germany, Sep. 2015, pp. 4197–4203.

[8] Z. Li, X. Zhou, W. M. Wang, G. Huang, Z. Tian, and S. Huang, "An ontology-based product design framework for manufacturability verification and knowledge reuse," *Intl. Journal of Advanced Manufacturing Technology*, vol. 99, no. 9, pp. 2121–2135, Dec. 2018.

[9] A. Perzylo, N. Somani, S. Profanter, I. Kessler, M. Rickert, and A. Knoll, "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Daejeon, Republic of Korea, Oct. 2016, pp. 2293–2300.

[10] A. Perzylo, S. Profanter, M. Rickert, and A. Knoll, "OPC UA nodeset ontologies as a pillar of representing semantic digital twins of manufacturing resources," in *Proc. of the IEEE Intl. Conf. on Emerging Technologies And Factory Automation*, Zaragoza, Spain, Sep. 2019, pp. 1085–1092.

[11] E. A. Topp, M. Stenmark, A. Ganslandt, A. Svensson, M. Haage, and J. Malec, "Ontology-based knowledge representation for increased skill reusability in industrial robots," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Madrid, Spain, Oct. 2018, pp. 5672–5678.

[12] M. Stenmark and J. Malec, "Knowledge-based instruction of manipulation tasks for industrial robotics," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 56–67, Jun. 2015.

[13] M. H. Arbo, Y. Pane, E. Aertbeliën, and W. Decré, "A system architecture for constraint-based robotic assembly with cad information," in *Proc. of the IEEE Intl. Conf. on Automation Science and Engineering*, Munich, Germany, Aug. 2018, pp. 690–696.

[14] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567–572, Aug. 2015.

[15] E. Kharlamov, N. Solomakhina, Ö. L. Özçep, D. Zheleznyakov, T. Hubauer *et al.*, "How semantic technologies can enhance data access at Siemens Energy," in *The Semantic Web – ISWC 2014*, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock *et al.*, Eds. Cham: Springer Intl. Publishing, Oct. 2014, pp. 601–619.

[16] E. Kharlamov, B. C. Grau, E. Jiménez-Ruiz, S. Lamparter, G. Mehdi *et al.*, "Capturing industrial information models with ontologies and constraints," in *The Semantic Web – ISWC 2016*, P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch *et al.*, Eds. Cham: Springer Intl. Publishing, Oct. 2016, pp. 325–343.

[17] L. Horvath and I. J. Rudas, "New approach to knowledge intensive product modeling in PLM systems," in *Proc. of the IEEE Intl. Conf. on Systems, Man and Cybernetics*, Montreal, Canada, Oct. 2007, pp. 668–673.

[18] S. Wang, J. Wan, D. Li, and C. Liu, "Knowledge reasoning with semantic data for real-time data processing in smart factory," *Sensors*, vol. 18, no. 2, Feb. 2018.

[19] M. del Mar Roldán-García, J. García-Nieto, A. Maté, J. Trujillo, and J. F. Aldana-Montes, "Ontology-driven approach for KPI meta-modelling, selection and reasoning," *Intl. Journal of Information Management*, vol. 102018, Oct. 2019.

[20] F. Pasic, B. Wohlers, and M. Becker, "Towards a KPI-based ontology for condition monitoring of automation systems," in *Proc. of the IEEE Intl. Conf. on Emerging Technologies and Factory Automation*, Zaragoza, Spain, Sep. 2019, pp. 1282–1285.

[21] C. Diamantini, D. Potena, E. Storti, and H. Zhang, "An ontology-based data exploration tool for key performance indicators," in *Proc. of the OTM Confederated Intl. Conf.*, Amantea, Italy, 2014, pp. 727–744.

[22] E. Järvenpää, N. Siltala, O. Hylli, and M. Lanz, "The development of an ontology for describing the capabilities of manufacturing resources," *Journal of Intelligent Manufacturing*, vol. 30, no. 2, pp. 959–978, Feb. 2019.

[23] B. Parsia, P. Patel-Schneider, M. Krötzsch, P. Hitzler, and S. Rudolph, "OWL 2 web ontology language primer (second edition)," W3C, W3C Recommendation, Dec. 2012, http://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[24] B. C. Grau, A. Fokoue, I. Horrocks, Z. Wu, and B. Motik, "OWL 2 web ontology language profiles (second edition)," W3C, W3C Recommendation, Dec. 2012, http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/.

[25] H. Knublauch, "SPIN - SPARQL syntax," W3C, W3C Submission, Feb. 2011, https://www.w3.org/Submission/2011/SUBM-spin-sparql-20110222/.

[26] S. Steyskal, D. Allemang, and H. Knublauch, "SHACL advanced features," W3C, W3C Note, Jun. 2017, https://www.w3.org/TR/2017/NOTE-shacl-af-20170608/.

[27] A. Seaborne, "SPARQL 1.1 query results JSON format," W3C, W3C Recommendation, Mar. 2013, http://www.w3.org/TR/2013/REC-sparql11-results-json-20130321/.

[28] G. Carothers and A. Seaborne, "RDF 1.1 n-triples," W3C, W3C Recommendation, Feb. 2014, http://www.w3.org/TR/2014/REC-n-triples-20140225/.