Multi-Level Modeling Framework for Machine as a Service Applications Based on Product Process Resource Models

Christian Brecher
Werkzeugmaschinenlabor, RWTH Aachen, Germany
c.brecher@wzl.rwth-aachen.de

Evgeny Kusmenko
Software Engineering (SE)Computer Science 3, RWTH Aachen,Germany
kusmenko@serwth.de

Achim Lindt
Software Engineering (SE)Computer Science 3, RWTH Aachen,Germany
lindt@se-rwth.de

Bernhard Rumpe
Software Engineering (SE)Computer Science 3, RWTH Aachen,Germany
rumpe@se-rwth.de

Simon Storms
Werkzeugmaschinenlabor, RWTH Aachen, Germany
s.storms@wzl.rwth-aachen.de

Stephan Wein
Werkzeugmaschinenlabor, RWTH Aachen, Germany
s.wein@wzl.rwth-aachen.de

Michael von Wenckstern
Software Engineering (SE)Computer Science 3, RWTH Aachen,Germany
vonwenckstern@se-rwth.de

Andreas Wortmann
Software Engineering (SE)Computer Science 3, RWTH Aachen,Germany
wortmann@se-rwth.de

ABSTRACT

At present, manufacturing processes are highly tailored to a specific product. Changes in product requirements therefore lead to big manual efforts for adapting the manufacturing process and reconfiguring production resources accordingly. Existing approaches do not cope well with this complexity. This hinders agile, customer-oriented manufacturing. A promising approach for automated assembling processes is the Machine as a Service paradigm, which aims for providing production resources on demand. This requires a consistent and pervasive formalization of product specifications, the corresponding manufacturing resources and their interdependencies. Thus, our first contribution is a generic and extensible multi-level and modular modeling framework to formalize products and available resources. Our framework is scalable for large companies and enables reuse for cross-company collaboration and supplier integration. Thereby, the static relationship between product, process and resource is avoided by describing product features and resource skills in separate models. Our framework uses the standardized SysML/UML. Our second contribution is the ability of our framework to integrate different standards. For demonstration, we apply our multi-level approach to a flexible assembly of terminal boxes for transmission gears and show the integration of standards by embedding the eCl@ss classification.

CCS Concepts

• Social and professional topics~Automation • Computing methodologies~Modeling methodologies • Hardware~Design for manufacturability • Hardware~Modeling and parameter extraction **• Software and its engineering~Model-driven software engineering • Software and its engineering~System modeling languages** • Software and its engineering~Abstraction, modeling and modularity • Software and its engineering~Unified Modeling Language (UML) • Applied computing~Industry and manufacturing

Keywords

Multi-level Modeling; Industry 4.0 Components; Asset Administration Shell; Integration of Ontologies; Language Adaption and Aggregation; Machine as a Service; PPR Models;

ISCSIC '18, September 21–23, 2018, Stockholm, Sweden

ACM ISBN 978-1-4503-6628-1/18/09…$15.00
https://doi.org/10.1145/3284557.3284714

1. INTRODUCTION

Globalization and world spanning logistics enable almost location-independent ordering, production and delivery of standard products. Due to the growing field of competitors, only the ability to quickly adapt to changing product demands and to enable a short time to market will yield competitive advantages in the future. This calls for an agile engineering approach that enables manufacturing customized, sometimes even individual products, facilitates high customer involvement and allows short ramp-up times. Emerging technologies related to the smart Cyber-Physical Production Systems [1] of Industry 4.0 can empower agility. The Machine as a Service (MaaS) paradigm describes one approach. The vision is a continuation of MaaS towards cloud connected production sites that can automatically analyze uploaded product models, match the specification with available resources, calculate and execute resource reconfigurations and ultimately derive and execute an automated production process. In addition, these production sites can expose their respective resource configuration model to allow for automated service discovery.

However, the complexity of (re)designing, implementing and deploying manufacturing processes according to changed product requirements is not well supported by existing modeling approaches and engineering tools. Hence, manual adaption of the manufacturing process requires great efforts, requires expert knowledge, experience and is error-prone. At present, production processes are mostly tailored to a specific but static product specification and are therefore inflexible for product variants or evolution.

Modeling the primary characteristics of products and resources requires structural specification techniques to define, for instance, product components and their assembly. CAD models offer precise means to this end and are therefore commonly used, especially since 3D printing became commodity. Nevertheless, CAD lacks support for modularity; i.e., syntactic and semantic integration of decomposed models. Thus, these models either are independently and scattered or monolithic and convoluted. Therefore, they are hardly comprehensible and inconvenient for modeling complex products with a variety of assembly features and complicate model reuse. Moreover, CAD models cannot be semantically integrated with the available resource specifications and process models.

To close these gaps between product specification, available resource skills and executable production processes, this contribution proposes a generic and extensible multi-level modeling [2] framework based on meta-models and a comprehensive formalization. The modeling elements of this framework are based on well-known and standardized SysML/UML notation. Additionally, the presented framework allows a seamless integration of existing ontologies and standards such as ISO, DIN, and eCl@ss classifications. The models are used to formalize and match product specifications with manufacturing capabilities and thus checking, whether and how available resources can produce a given product. The approach counteracts complexity by leveraging hierarchical abstraction levels and composability of models. This enables scalability and makes the approach feasible for large projects consisting of many development teams. Its modularity ensures a clear assignment of responsibilities and roles providing a tailored view for each stakeholder thereby hiding unnecessary complexity while enabling re-usability for cross-company collaboration and supplier integration. The multilevel-modeling framework serves as a foundation towards Production as a Service and future work towards automated derivation of production processes.

The feasibility of the proposed approach has been evaluated using the example of a flexible assembly of terminal boxes for transmission gears by integrating the eCl@ss classification of screws.

The rest of the contribution is structured as follows: The following section gives a brief introduction to Machine as a Service and to related approaches. Section 3 explains the proposed approach by introducing the meta-level structure of the proposed modeling framework as well as a product and resource meta-model. The presented approach is elucidated in section 4 by an exemplary application. Section 5 gives a conclusion and a brief outlook for future work.

2. PRELIMINARIES

This section summarizes works of other groups in the context of Industry 4.0, product process resources and machine as a service on which our approach for the later presented modeling framework builds on.

2.1 Machine as a Service/Manufacturing as a Service

Machine as a Service and Manufacturing as a Service are two important concepts in the context of production networks, Big Cloud Fabric 4.0 or Industry 4.0. Enabled by developments like Internet of Things and service-oriented manufacturing platforms, machines become part of service controlled global networks of manufacturing systems [3]. The Industrial Internet of Things allows software rollout for controlling production processes on every machine needed for production. These software updates allow adapting manufacturing tasks based on virtual product model, so that in future, costumers just configure their products, which will be produced in a worldwide production network. Yet, a profound estimation of the required resources for production processes of virtual product models is still difficult.

An approach for defining resources for service oriented cyber-physical manufacturing systems (CPMS) [1] focuses on modeling resource capabilities as these are key factor for service matching. Therefore, CPMS are categorized into processing, transporting and storage regarding pre-conditions, assumptions, post-conditions, and effects of the capabilities. The main categories and their subclasses are grouped in a CPMS taxonomy including aspects such as the range of workable material and the ability to meet product-manufacturing specifications. Even though, the presented method [1] describes a method for goal-service matching; a concrete definition of the relation between product and resource model is missing.

2.2 Product Process and Resource Modeling for Industry 4.0

Many standards such as AutomationML define the entities product, process and resource (PPR) and their relations in a triangle [4] similar to the one depicted in Figure 1. The disadvantage of combining all these entities via processes is that the product specification is now directly related to the process, and thus other manufacturing tools cannot easily reuse it.

While there already exists integrated platform models for products and manufacturing systems during the conceptual design phase [5], the product and the resource model are connected via processes and operations. The correlation between product and process for example can be carried out using a product definition model according to IEC 62264-1 [6].

Ferrer [7] presents an UML-based approach following this division by defining classes for product, system, station and component as well as operation, process and task, which are specified by object properties. However, essential information on production technology is missing; e.g., the processing sequence of production processes cannot be displayed and the mapping of a product to the necessary processes and resources, depending on the individual properties of the product, cannot be performed.

2.3 Industry 4.0 Components

The Industry 4.0 Component (I4.0Comp) presents a specific case of Cyber-Physical Systems (CPS). The I4.0Comp of a physical or non-physical entity consists of a real asset; e.g., a proximity sensor, and its digital representation. This representation is deposited in an asset administration shell (AAS), providing multiple sub models and property value statements for description data, like functionalities, geometric structure, positioning, etc. These sub models and properties can be originated from different sources, allowing suppliers and manufacturers to provide type information or even first instance information about their products to the customer. Besides, the asset's life cycle data can also be tracked and deposited in the respective AAS [8].

Grangel-Gonzáles [9] presents an approach on semantic formalization and modeling of I4.0Comp utilizing the Resource Description Framework (RDF). Thereby, also the integration of standards and vocabularies, like IEC 62264, eCl@ss and Ontology of Units of Measure, into I4.0Comp is considered. Yet, a method

for integrating AASs in product and resource modeling is still missing.

2.4 Ontologies

Ontologies emerged in the field of factory automation to reduce the amount of, e.g., possible assembly features by establishing firstly categories and a hierarchical order and secondly by defining possible attributes for each level of detail; examples are the classification of manufacturing processes as in DIN 8580 [10] and assembly methods described in DIN 8593 [11].

In contrast to UML class diagrams, ontologies do not support templates or composite structures, which are useful for predefining incomplete assembly groups in modular systems [12].

2.5 Ontologies for Production Systems

The P-PSO (Politecnico di Milano-Production Systems Ontology) approach for ontology-based modeling of manufacturing systems [13] addresses three different aspects for modeling a manufacturing system: (i) the physical aspect is concerned with the material definition of the system - e.g., workers, material handling, tools; (ii) the technological aspect focuses on the functional process; and (iii) the control aspect regards the overall planning and scheduling. The technological aspect contains concrete modeled transformations that the product parts must undergo within the manufacturing system in a concrete order; this makes the manufacturing process highly dependent to one specific production system.

To support multiple compositions of components the P-PSO utilizes ontologies to specify the object models of different component classes. In addition to the manufacturing system, products can be modeled as parts with variants, while processes are modeled as operation-process charts with an object-oriented operation class.

The relations between the system components, the process operations as well as the product parts are modeled primary statically using UML class diagram relations.

This leads to a partly detailed model of the PPR system - as can be seen for logistic systems [14]. By extending the PPR model with the additional classes, skills and tasks, the interdependences between the main components of the model are formally described [4]. For this purpose, a skill defines the ability of a resource to perform processes, whereas the application of a skill on a specific product type is called task. Skills as well as tasks have a certain set of properties, which contains additional information.

An ontology based matching algorithm [15] for resources and processes is based on the modeled capabilities. These capabilities are defined as the resources' ability to perform specific tasks. The task indirectly refers to the product model. Due to a missing specific product model, the direct comparison of the product and the production system model cannot be done.

2.6 Multi-Level Modeling

One important purpose of domain models is to serve as a description of the problem; its main aim is to determine whether an agreement about the system requirements has been achieved and whether the system domain has been accurately captured [2]. By definition, the overriding goal of domain modeling must be to represent the problem space concepts in as faithful and untarnished a way as possible. In addition, most domains, such as the production domain, have more than just the two UML level types. Considering a connecting element on the highest abstraction level. One level lower and more specifically it becomes a screw, and in even more concrete levels inner screw and outer screw, and eventually an instance of an inner screw is a metal threaded screw with thread M2.

3. APPROACH

Based on the existing definitions for product, process and resource relations, e.g., an adapted PPR model is proposed [16] as shown in Figure 1. By defining product requirements and resource skills the fixed relation between product, process and resource can be dissolved, which is described in the following.

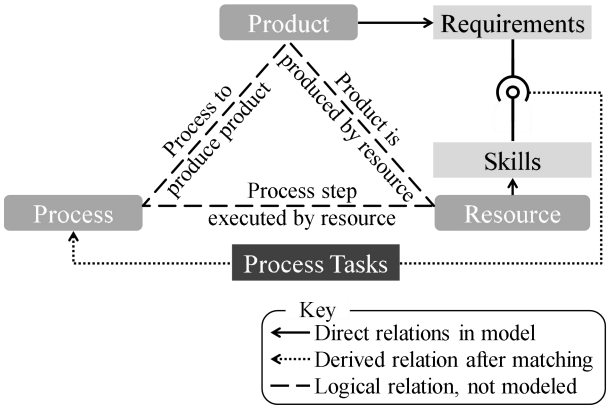

Figure 1. Adapted Product Process Resource Model [16] to have no direct relationship between resource and product.

To tackle the difficulties imposed by the aforementioned methods, we present a novel versatile multi-level modeling framework for manufacturing. The notion of multi-level modeling, originally emerging from the field of software engineering, is a key element in the design of large systems. Its aim is the assembly of a product from generic building blocks, which are steadily refined until the concrete realization level is reached, and the concrete product instance can be assembled. Such a hierarchical approach has a series of advantages compared to a flat modeling methodology as pure ontology-based modeling presented above. It enables an agile development process allowing dedicated teams to work on different parts of the system, which can be assembled seamlessly due to clear interfaces prescribed by the abstract high-level models. Even more, the modeling of aspects an OEM might not be interested in can be underspecified and delegated to suppliers offering the required domain knowledge. The responsible team in turn need to deliver their parts in accordance to the imposed interfaces and constraints.

The structure of the proposed meta-modeling framework is inspired by MOF, a meta-modeling framework for the domain of object-oriented systems [17]. It is organized in the four levels L0 to L3 depicted in Figure 2 ranging from most abstract (L0) to concrete (L3). As is inherent for multi-level modeling, each abstraction level is an instance of its predecessor. In contrast to classical paradigms such as object-oriented programming where a class must be instantiated completely at one go, instantiation is performed lazily in multi-level modeling. Thus, an instance residing at an intermediate level may still contain abstract parts, which need to be concretized at one of the successive levels.

The goal of the model-based approach presented here is to specify

product requirements and resources abilities in such detail, that a product can be assembled by utilizing appropriate resources with matching skills. Hence, the approach starts by specifying the meta-

models for products as well as production plants in the scope of mechanical assembly processes. Therefore, we distinguish between product (meta-) models on the one hand side and resource (meta-) models on the other, both dealing with the notion of the physical aspect as defined in P-PSO.

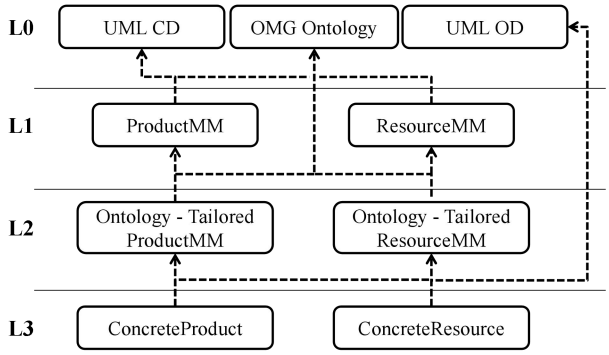

Figure 2. Manufacturing meta-modeling framework based on MOF [17].

The meta-meta-level (L0) specifies a set of modeling languages, employed by our framework. Thereby, we rely mainly on structural diagrams of the UML 2.0 specification, particularly class (CD) and object (OD) diagrams. Providing means of hierarchical decomposition, abstraction/specialization relationships, etc. these languages cover the needs of manufacturing modeling. On the other hand, ontologies are widely used in mechanical engineering to formalize international standards such as ISO, DIN, and others by providing detailed properties for specific manufacturing tasks and by categorizing parts. Therefore, we integrate ontology languages into our framework. Fortunately, the hierarchical concept of ontologies fits very well to the inheritance concept of class diagrams allowing us to integrate the ontology models with the product models in the following levels.

The gap from the high-level modeling language concepts to concrete product and plant models (L3) is filled by two intermediate modeling levels L1 and L2: The meta-model level L1 specifies the general structure of arbitrary products and resources: their respective components, their interactions, relationships, and properties. This level provides a fixed modeling language for the domain of manufacturing, which is used by the engineers to design abstract models of arbitrary products and resources, respectively.

3.1 The Product Meta-Model

The elements of the product meta-model are depicted in Figure 3. Generally, assembly processes deal with physical product components, which are being assembled step by step. As is denoted by the diamond notation of UML, a *Product* is composed of *StructuralElement*s. The proposed product meta-model is described in the following.

We distinguish between two kinds of *StructuralElements*: *Components* are either an atomic part, which in turn can be a *Part*; e.g., the lid of a gearbox or a *ConnectingElement; e.g.,* a bolt, a screw, or a weld. Components may be subdivided in physically connected or subcomponents. Secondly, we introduce the notion of *Modules* grouping multiple *Components* or further modules to form a new *StructuralElement* thereby allowing the definition of hierarchies of logical submodules facilitating reuse. All structural elements may have corresponding 3D CAD model and are interconnected via two possible types of linkages: assemblies and alignments. An *AssemblyFeature* is the generic abstract super class for various assembling of product components (e.g., form-fit, force-fit etc.). This class serves as the main extension point for ontology-based specifications and must be refined in Level L2 with a concrete assembling strategy and according parameters. An *AssemblyFeatureGroup* allows the definition of logical clusters of single *AssemblyFeatures*, which may require a holistic process step; e.g., the definition of a screw group indicates the need for a specific screw pattern to avoid tilting. Each *AssemblyFeature* requires the specification of an alignment between the structural elements involved. Further alignments may be specified at any time without a direct need of an assembly feature. *AlignmentFeatures* precisely specify these alignments; e.g., the angular orientation of two metal sheets towards each other; e.g., co-axial, parallel, or perpendicular, for a scarf joint welding process. By utilizing *AssemblyFeatures* and *AlignmentFeatures* in combination with multiple parts and connecting elements, a whole assembly can be described. An additional *FabricationFeature* refers to processing steps, which affect components, modules or the complete products such as shape cutting or finishing. Each of the classes *AssemblyFeature*, *AlignmentFeature*, *FabricationFeature* and *ConnectingElement* forms an extension point for ontology-based refinements.

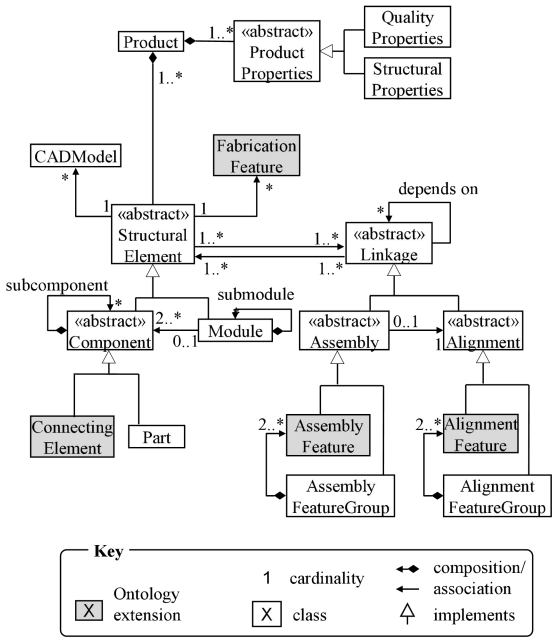

Figure 3. Product meta-model (L1).

3.2 The Resource Meta-Model

By addressing the whole automation pyramid, the resource hierarchy defined in IEC 62264-1 allows the information escalation from work cells over production sites up to the Enterprise Resource Planning (ERP) system while leaving out a sophisticated approach on the utilized technical equipment. For a semantical matching of the product requirements with the model of a production system, a concrete description of the system's capabilities is necessary. Therefore, the hierarchy presented in IEC 62264-1 must be extended by a meta-model that specifies the properties of a work cell and its resources (L1). The resource meta-model pictured in Figure 4 is a more concise advancement of the hierarchical ontological model class diagram of product, process, and resource introduced by Ferrer [7].

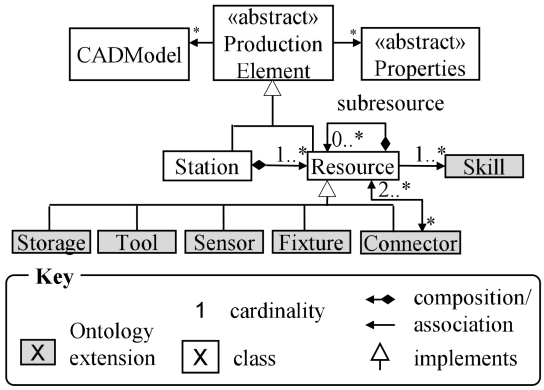

Figure 4. Resource Meta-Model (L1).

In the presented approach, a production site consists of one to multiple stations. Each *Station* contains at least one *Resource* that can have and utilize sub resources. For example, the resource 'robot' can use a connected sub resource 'gripper' to manipulate an object while measuring the object's geometry with an attached 'camera'. In this case, the resource robot is a complex resource with depending sub resources. Stations as well as resources are both of type *ProductionElement* and thereby have *Properties* and an assigned *CAD Model*. The eCl@ss standard works as a basis for the definition of station and resource properties. Additional properties can be attached to the model at any time. The concrete resource *Tool* is defined as a resource that can directly interact with or modify a product.

A *Sensor* is always coupled with its underlying data processing. Beside tools and sensors, Resources can be linked both physically and logically using a *Connector*. By connecting resources of different stations, the stations are logically linked. Physical assets for these Connectors are; e.g., robots or conveyor belts. Following the P-PSO [13] approach (see II-D), additional specialized resource sub-classes *Storage* and *Fixture* are integrated in the resource meta-model, as they are frequently used resources in industrial assembly. The functional capabilities of the station are modeled as *Skill* for every resource. Based on the extended PPRS approach [4], skills are defined as the technical ability of a resource to perform processes. Beside this relation of a process and resources, each skill has a specific set of process relevant properties. For example, example a *Tool* 'Screwdriver' has the skill that allows the tool to tight socket head screws with a certain kind of tightening method and within a specific range of tightening torque. The property set for each skill is derived from the deposited resource properties. By analogy to the resource hierarchy, skills of complex resources are derived from the respective skills of the available sub resources. For example, the complex resource robot has the skill positioning in X, Y, Z-axis from the robot and additionally the skill part manipulation from the attached gripper. Like the assembly features in the product model, resources and their skills cannot be expressed on this level, due to the high variety of resources in the industry.

To improve the model's level of detail in preparation of the process derivation, the class *Properties* can be derived to express specific additional properties like Status, Accessibility or Energy consumption for stations and resources. This also enables a registration of the machines availability across different production sites. Beside this functional model, the geometries of the stations and resources are deposited in the CAD model for later process relevant simulation aspects.

3.3 Modeling Levels

Having the meta-model defined on level L1 at one's disposal, engineers can now use it to create specific product and resource models. As stressed in the sections above, concrete models are not created directly. Instead, the modeling process is subdivided into two intermediate levels. On level L2, *abstract manufacturing models* are defined. In Figure 5, we demonstrate on two simple examples how the meta-models can be employed to describe arbitrary assembly models. A basic screw connection as depicted on the left-hand side consists of the parts to be connected, each being an instance of the *Part* class defined in Figure 5 and two *ConnectingElements*, namely the screw itself and a nut. A completely different type of assembly is the plug connection of two Duplo bricks depicted on the right-hand side of Figure 5. The Duplo bricks are *Part* instances of our modeling framework whereas their plugs and sockets need to be modeled as *ConnectionElements.*

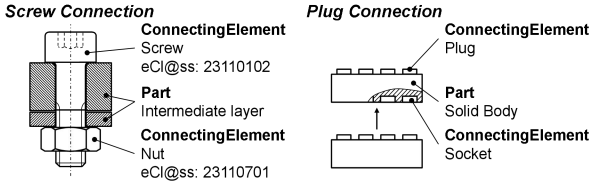

Figure 5. Assembly examples of a screw and a plug connection to be modeled using the multi-level modeling framework.

A major advantage of the multi-level modeling approach, particularly in comparison with P-PSO only using standard UML modeling, is that the models on level L2 do not have to exhibit all information needed to realize an assembly. Entities on this level are an equivalent to the concept of [2], a hybrid of class and object, in object-oriented programming; i.e., they are partially instantiated classes. Many properties such as the color of a part, concrete material variant, or a specific screw realization are not relevant for the understanding of the overall assembly and may remain underspecified. This has multiple advantages: often details need to be handled by a specialist or a supplier and should be hidden from the management or the OEM. On the other hand, the OEM can introduce constraints by providing ranges for the concrete realization; e.g., for the diameter of a bolt. The supplier then needs to deliver a realization fulfilling the range requirement while having the flexibility to choose the most favorable manufacturing variant.

In our framework, the technological aspects of P-PSO are included in the definition of product features and on the resource side within the skill definition. In contrast to P-PSO, the technological aspect is thus not hard-wired in our concept allowing flexible and adaptable production processes. Therefore, we forgo task or process modeling and only focus on requirement and skill modeling instead which is a major achievement. In contrast to current digital descriptions of assembly systems, where the product is tightly linked to the processes, skills, and resources, we aim at a loose coupling where a product is only associated with the skills corresponding to its requirements as depicted in Figure 1. This in turn allows an automated and flexible mapping onto processes and available resources and is particularly important for MaaS platforms where resources can be allocated dynamically while aiming at working to capacity. Our framework supports the control process used in P-PSO as well; e.g., to ensure the amount of resources needed based on the skills. However, this aspect is out of scope of this paper.

To ensure compatibility of the terminology and parts used, manufacturing processes heavily rely on standards. As mentioned above, our framework provides ontology languages on L0 making it possible to use existing ontologies describing concrete standards on L1, similarly to P-PSO. Briefly, ontologies provide efficient means for structuring and querying arbitrary information in semantic networks. Particularly, OWL provides a powerful way for accessing and querying ontologies that cannot be applied to UML class diagrams. The approach presented here does not aim to embed ontologies completely into the meta-models defined in L1. The proposed methodology is to query appropriate ontologies and then convert the essential classes and properties into a corresponding UML class with according attributes.

To ensure a seamless integration of the referenced ontology elements into the product and resource models, a bridge from the technological space of ontologies to the technological space of UML class diagrams is required [18]. The bridge maps the ontology, usually provided in the XML format, to our class diagram language provided by L0. This mapping enables a direct usage of the entities defined in the ontology by the product and resource models available in the class diagram format as well. Therefore, to include ontology names of a standard into product or resource models in a semantically correct way, the product or resource models must be able to resolve the names and their properties defined in the standards. A resolution denotes a name-based search for an element inside the model. Since after the bridging step the standards reside in a separate model of the technological space of UML class diagrams, it can be adapted to the product and the resource model, respectively. The adapter forwards the resolution requests from the product and the resolution models to the corresponding standards models and returns information about the required standard as if it was available directly in the model looking for it. This type of model composition is implemented using the concept of language aggregation provided by the several language workbenches such as MPS [19], MontiCore [20], or GEMOC Studio [21].

For screw classification, we employ the widely used eCl@ss [22] taxonomy. Note, that on the abstract model level L2, one does not need to constrain the screw to a specific eCl@ss entry. Instead, sets or ranges of possible values can be specified. The tailored model is depicted in Figure 6. A screw defined in level L2 of the product model resolves the screw name in the corresponding ontology-based model specifying its eCl@ss ID and hence, decoupling the product model from the concrete realization. Note that capacities (c.f. P-PSO capacities) such as throughput or maintenance intervals are declared as properties of ontology-tailored models, as well.

In contrast to L2, models on L3 must be free of any underspecification. Properties must be set to concrete values fulfilling the imposed constraints but not leaving any freedom for interpretation, i.e. such that each entity in the product model can be mapped to a concrete part number.

The obtained model of the screw connection allows an automated verification of the desired skills. What is more, due to the leveled instantiation process not all the skills need to be verified at one go, i.e. not all the required resources need to be assigned immediately but can be postponed for an assignment at an appropriate point in time, possibly by another party. For instance, at level L2 the requirement to realize a screw-based connection expects an assembly tool for screws and nuts. The specification of numeric properties including the tightening torque, diameter, and drive may and should be delayed to L3 modeling. Based on the requirement models, possible suppliers can be chosen by verifying whether their respective skills fulfil the assembly requirements.

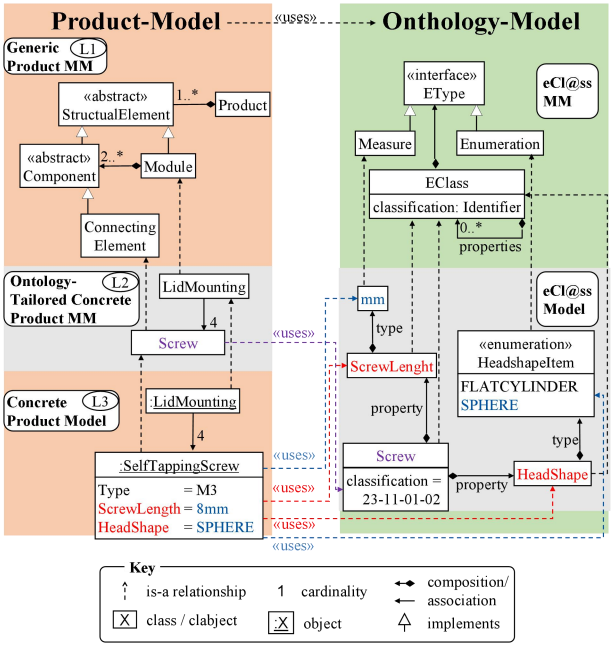

Figure 6. Tailoring a product model with an ontology.

Once a final concrete model is available, the questions arises, how variants of this model can be derived and managed. Due to their modular nature, class diagrams are easy to extend and adapt to obtain a desired variant. Multi-level modeling however leverages this modularity to a new level: a model can be adapted by changing instantiations on each level [2]. The complexity of possibly emerging variants can be tamed; e.g., by using UML Feature Models. However, the aspect of variant management is out of scope of this paper.

3.4 Matching Product Requirements with Resource Skills

The method presented in this paper, namely directly linking product models with resource models shows a novel approach for matching product specifications with available resources in an industrial context. The analysis and comparison use L2 and L3 models with integrated ontologies of each, product and resource. Enabled by the ontologically refined assembly and fabrication features of the product model as well as the skills of the resources model a direct semantic comparison can be made. Therefore, the features are formalized as product requirements on the resource. Requirements may be a combination or split of features defined for the L2 product model. Additionally, various skills can be combined to super ordinated skills. This process is called resolving namespace. The comparing progress can be split into two steps. First, the matching resources – able to manufacture a specific product feature – are identified by selecting the semantic description of a requirement und comparing it with the skill descriptions of all available resources. If the namespace deposed in the skill class of a resource model corresponds to a requirement description, the specific resource is marked and thereby allowed for the next analysis step. In the second step, the properties of both, product requirements and linked skills of the marked resource are numerically matched. A resource can produce a specific feature, if the skill properties of the resource match the properties of the

product requirement. It is also possible that more than one resource matches the requirements of a product model. In this case, both resources are linked to the product feature. The final selection of the resource to produce the product feature is made in line with the process scheduling.

3.5 Utilization of Asset Administration Shells in Modeling

The proposed framework can be improved by utilizing AASs for specific Structural-Elements or Resources. As supplier and manufacturer are enabled to provide their concrete models via internet, standard parts and their respective properties can be integrated directly in the modeling of products and resources. Thereby, live-updates of component descriptions and detailed component information can be shared.

4. EXEMPLARY APPLICATION

The modeling framework is evaluated in an exemplary use case, setting up the UML-based L2 and L3 models for a specific product and a specific resource: A terminal box for transmission gears assembled in a flexible, reconfigurable robot assembly cell. Different lids have to be assembled on a terminal box using either gripping or screw driving skills.

4.1 General Set-Up

The terminal box has four cable ducts, which are individually closeable, depending on the customers' requirements. While two ducts are closed with lids, having a radial seal as well as an integrated thread and slot drive, the other two ducts can be closed with a lid using four self-tapping screws, mounted in through bores. Bores in the body of the terminal box serve as counterpart for the screws, having no thread before assembly. Besides, an additional rubber seal must be applied to a groove in the cab, sealing the inside of the box against moisture. For the case study, we focus on the second assembly. The assembly set-up is shown in Figure 7.

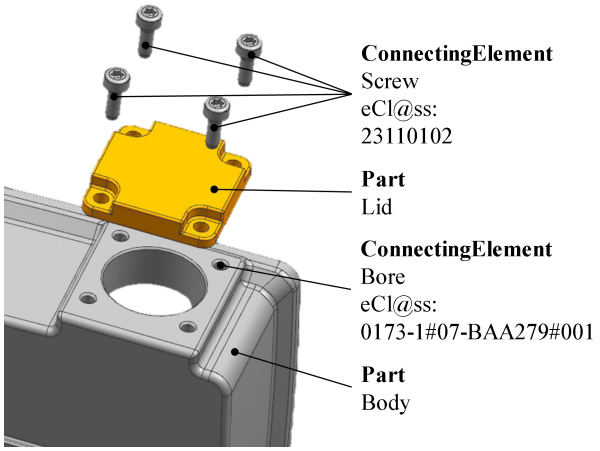

Figure 7. Exemplary assembly set-up.

The assembly cell consists of a 6-axis articulated robot, which can manipulate different tools. The tools can be automatically exchanged during the assembly process utilizing an integrated tool interface, making the assembly cell reconfigurable. The tools integrated in the assembly cell are two parallel grippers and a screwdriver. Additionally, one of the grippers has an interface to change between different jaws depending on the needed grip variant; e.g., outside diameter or intermediate gripping. Therefore, a storage for the jaws is integrated into the assembly cell. In addition, the socket of the screwdriver is automatically changeable. A socket driver set provides the corresponding socket drives. To fix any object during the assembly process, a flexible clamping system with two claws is integrated. Beside the actuators, the cell includes two cameras for the direct process support. One of the cameras is fixed at the roof of the cell, to determine the position of objects in the working space. The other camera is attached to the robot, which is used for online position correction.

4.2 Exemplary Product and Resource Multi-Level Models

In the following, selected parts of the described product and resource are modeled using the L2 and L3 models. An excerpt of the models for product and resource based on the defined multi-level modeling approach is shown in Figure 8 and Figure 9.

4.2.1 Product

First, the ontology-tailored concrete product meta-model (L2 model) is generated. The product terminal box consists of the components solid Body, Screws, Bores and Lid. The screws and bores are modeled as ConnectionElements, whereas the lid as well as the body are modeled as Parts. Screws, bores and the lid form a module, named LidMounting. In preparation of modeling these logical modules, feature detection analyzes the assembly within the products CAD model, returning interdependencies between different assembly components [16]. In the given use case – lids fixed with self-tapping screws – the geometrical relation of the screws with the intermediate layer and join partner is analyzed. The derived relation is then modeled as an AlignmentFeature. In this case, the main axis of the screw, the bore of the intermediate layer and the join partner are coaxial to each other (not shown in Figure 8).

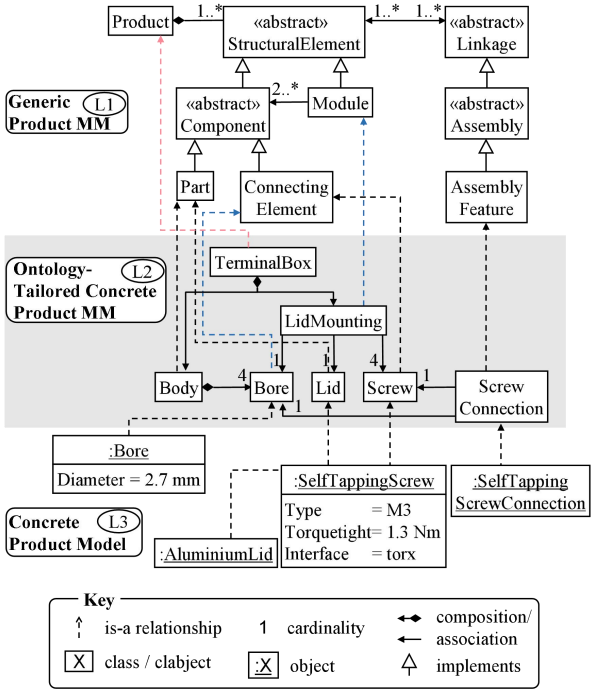

Figure 8. Excerpt of the exemplary product multi-level model.

Furthermore, the associated *AssemblyFeature* defines the connection type of the assembly. In this case, the assembly is realized by a screw connection, linking one screw with one bore of

the terminal box body. Additionally, the screw connections can optionally be grouped in an *AssemblyFeatureGroup*, allowing the integration of screw patterns, as an early specification of the dependencies between the four screw connections.

Second, the concrete product model (L3 model) is specified, adding concrete numerical and semantic values as properties to the components as well as to the *AssemblyFeature ScrewConnection*, as shown in Figure 8. For example, the screws are specified as self-tapping screws of the type M3 with the drive interface torx and a tightening torque of T_{Tight} = 1.3 Nm.

4.2.2 Resource

For the resource model, a possibly existing CAD model is linked to the ontology-tailored concrete resource meta-model (L2 model) and then derived to a concrete resource model (L3 model). The assembly cell is modeled as a station with multiple resources. The major resource of the station is the 6-axis Robot, which is modeled as a complex resource, with different sub-resources: a tool interface and 2D camera. The robot itself has the skills *3DPositioning* und *ForceControl* due to integrated sensors. By mounting the different tools like gripper or screwdriver, additional skills are added. The *Screwdriver* enables the skill *ScrewDriving* and the *Gripper* allows *ObjectManipulation*. While socket drivers and gripping jaws are exchangeable, both, the screwdriver and the gripper can be modeled as complex resources, adding further skills to the resource model. Jaw storage, commission boxes and socket driver set are described as the resource sub class *Storage*, providing the skill store with multiple entities. Besides, other resources like the flexible clamping system, modeled as fixture, provides the skill Fixate. Figure 9 shows an excerpt of the described ontology-tailored concrete resource meta-model.

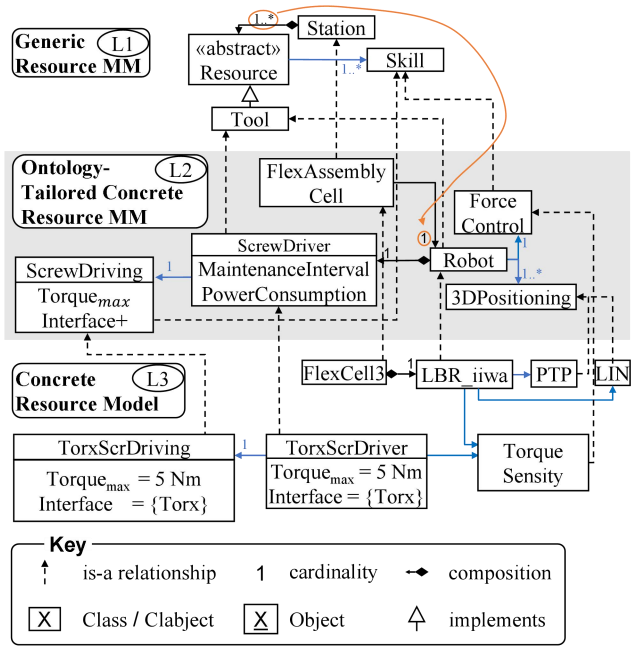

Figure 9. Excerpt of the exemplary resource multi-level model.

This model is specified within the concrete resource model. The *FlexCell3*, has the specific instance *LBR_iiwa* of the type *Robot*, which is capable of the skill *3DPositioning* with Point-to-Point (PTP) and linear (LIN) movements. Additionally, the sub-resource *ScrewDriver* is completely specified as the instance *TorxScrDriver* with filled properties; e.g., the range of the screwdrivers applicable torque is limited to a maximum torque of five newton meters. It enables the skill *TorxScrDriving* that is also fully specified.

As illustrated above, the model is gradually refined in each level of modeling, also adding more constrains. This allows the online modeling of complex models in cross-side teams and a preforming of the possible modeling. The multi-leveling allows mapping to the hierarchical structure of the domain and or organization, preventing failure in modeling.

5. CONCLUSION

The contribution of this work is twofold: (1) the formalization of a generic but modular and extensible product and resource meta-model, which allows to incrementally concretizing the abstract meta-models towards specific product variants and resource specifications; and (2) the possibility to incorporate existing modeling and knowledge resources from industry standards and ontologies. The leveled approach minimizes complexity and enables model reuse instead of dealing with monolithic and therefore incomprehensive and overloaded models. This may encourage establishing and even possibly sharing reusable product component and resource libraries in the future.

The multi-level modeling framework provides a foundation and another step towards Production as a Service platforms emerging in the context of Industry 4.0 and existing Machine as a Service approaches. It would be highly desirable that these platforms will be capable of automatically discovering matching production resources to a provided product specification. The assembly features defined in the product model, along with their optional dependencies linked with matching resource skills seem to form feasible building blocks for deriving process steps, which can eventually be chained to an executable process model. As a concluding step towards agile manufacturing, we will prospectively focus on a methodology to derive concrete and deployable manufacturing processes.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of the Profile Area Information and Communication Technology of the RWTH Aachen and of the German Research Foundation (DFG) through the Cluster of Excellence “Integrative Production Technology for High-Wage Countries”.

REFERENCES

[1] Y. Lu and F. Ju, “Smart Manufacturing Systems based on Cyber-physical Manufacturing Services (CPMS),” IFAC-PapersOnLine, vol. 50, no. 1, pp. 15883–15889, 2017.

[2] C. Atkinson and T. Kühne, “Reducing accidental compleity in domain models,” *Softw Syst Model*, vol. 7, no. 3, pp. 345–359, 2008.

[3] F. Tao and Q. Qi, “New IT Driven Service-Oriented Smart Manufacturing: Framework and Characteristics,” *IEEE Trans. Syst. Man Cybern, Syst.*, pp. 1–11, 2017.

[4] J. Pfrommer, M. Schleipen, and J. Beyerer, “PPRS: Production skills and their relation to product, process, and resource,” in *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), 2013: 10 - 13 Sept. 2013, Cagliari, Italy*, Cagliari, Italy, 2013, pp. 1–4.

[5] G. Michalos *et al.*, “Autonomous Production Systems Using Open Architectures and Mobile Robotic Structures,” *Procedia CIRP*, vol. 28, pp. 119–124, 2015.

[6] ISO, *IEC 62264-1:2013: Enterprise-control system integration -- Part 1: Models and terminology.* [Online] Available: https://www.iso.org/standard/57308.html. Accessed on: May 30 2018.

[7] B. R. Ferrer *et al.,* "An approach for knowledge-driven product, process and resource mappings for assembly automation," in *2015 IEEE International Conference on Automation Science and Engineering (CASE): 24 - 28 Aug. 2015,* Gothenburg*, Sweden*, Gothenburg, Sweden, 2015, pp. 1104–1109.

[8] C. Wagner *et al.,* "The role of the Industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation: September 12-15, 2017, Limassol, Cyprus*, Limassol, 2017, pp. 1–8.

[9] I. Grangel-Gonzalez *et al.,* "An RDF-based approach for implementing industry 4.0 components with Administration Shells," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA): September 6-9, 2016 Berlin, Germany*, Berlin, Germany, 2016, pp. 1–8.

[10] DIN, *DIN 8580:2003-09: Manufacturing processes - Terms and* definitions*, division.* [Online] Available: https://www.beuth.de/de/norm/din-8580/65031153. Accessed on: May 30 2018.

[11] DIN, *DIN 8593-0:2003-09: Manufacturing processes joining - Part 0: General; Classification, subdivision, terms and* definitions*.* [Online] Available: https://www.beuth.de/de/norm/din-8593-0/65031206. Accessed on: May 30 2018.

[12] K. Kiko and C. Atkinson, *A Detailed Comparison of UML and OWL: Technical Report TR-2008-004*.

[13] M. Garetti and L. Fumagalli, "P-PSO ontology for manufacturing systems," *IFAC Proceedings Volumes*, vol. 45, no. 6, pp. 449–456, 2012.

[14] L. Fumagalli, S. Pala, M. Garetti, and E. Negri, "Ontology-Based Modeling of Manufacturing and Logistics Systems for a New MES Architecture," in *IFIP Advances in Information and Communication Technology*, vol. 438, *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World: IFIP WG 5.7 International Conference, APMS 2014, Ajaccio, France, September 20-24, 2014, Proceedings, Part I*, B. Grabot, B. Vallespir, S. Gomes, A. Bouras, and D. Kiritsis, Eds., Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg, 2014, pp. 192–200.

[15] A. Argyrou, C. Giannoulis, N. Papakostas, and G. Chryssolouris, "A Uniform Data Model for Representing Symbiotic Assembly Stations," *Procedia CIRP*, vol. 44, pp. 85–90, 2016.

[16] C. Brecher, S. Storms, C. Ecker, and M. Obdenbusch, "An Approach to Reduce Commissioning and Ramp-up time for Multi-variant Production in Automated Production Facilities," *Procedia CIRP*, vol. 51, pp. 128–133, 2016.

[17] OMG Object Managing Group, *Meta Object Facility (MOF) Core* Specification *2.5.1: Technical Report.* [Online] Available: https://www.omg.org/spec/MOF/About-MOF/. Accessed on: May 30 2018.

[18] T. Walter, *Bridging technological spaces: Towards the* combination *of model-driven engineering and ontology technologies*. Zugl.: Koblenz, Univ. Koblenz-Landau, Diss., 2011. Berlin: Logos-Verl., 2011.

[19] M. Voelter and V. Pech, "Language modularity with the MPS language workbench," in *34th International Conference on Software Engineering (ICSE), 2012*, Zurich, 2012, pp. 1449–1450.

[20] R. A. U. S. E. RWTH Aachen University Software Engineering Group, *The MontiCore 4.x Language Workbench: Technical Report - Aachen, December 12, 2017,* 1st ed. Herzogenrath: Shaker, 2017.

[21] M. Mazzara and B. Meyer, Eds., *Present and ulterior software engineering*. Cham: Springer, 2017.

[22] D. Fensel *et al.,* "Product data integration in B2B e-commerce," *IEEE Intell. Syst.*, vol. 16, no. 4, pp. 54–59, 2001.