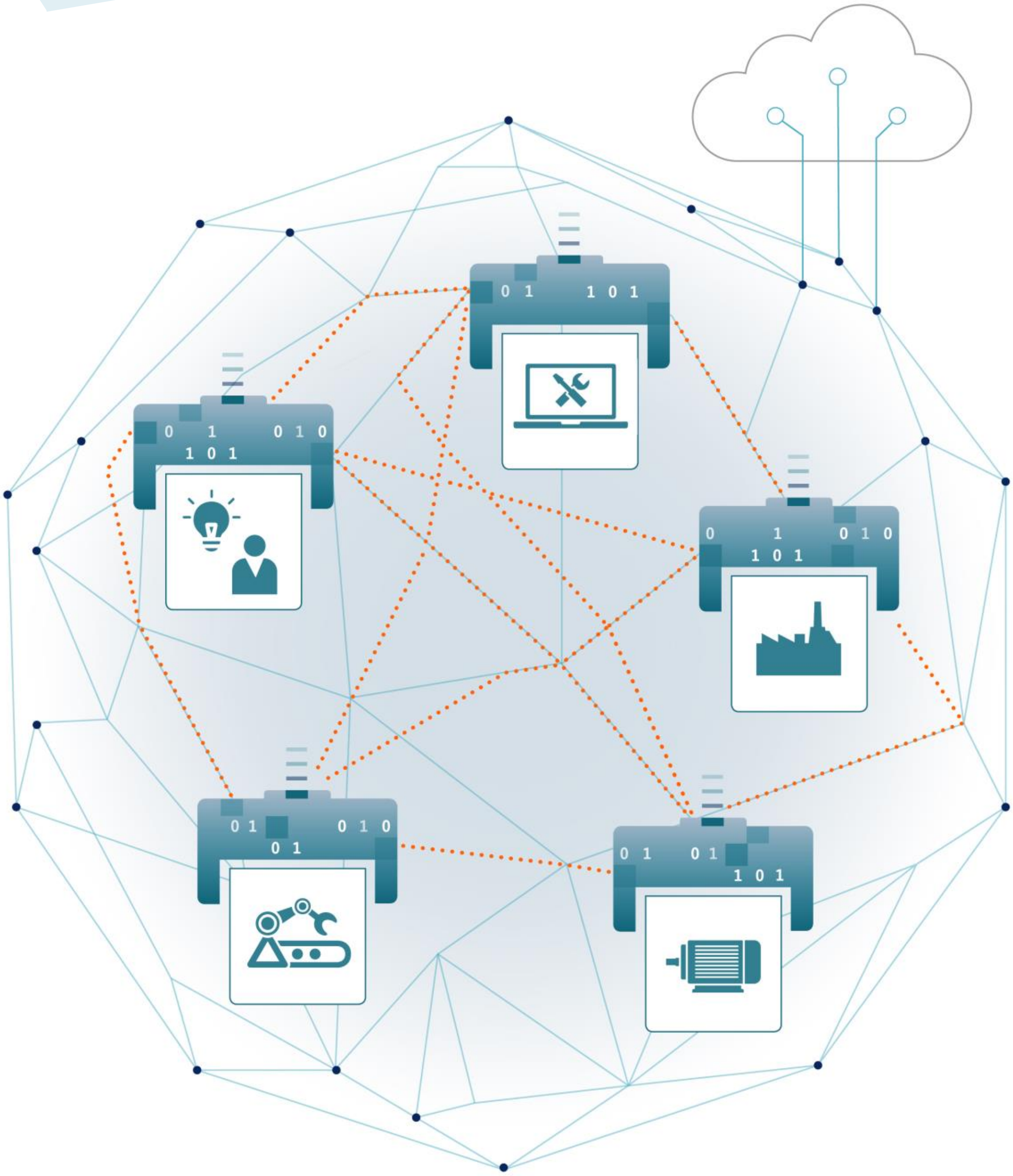
PLATTFORM INDUSTRIE 4.0

Discussion paper



Information Model for Capabilities, Skills & Services

Definition of terminology and proposal for a technology-independent information model for capabilities and skills in flexible manufacturing

Imprint

This document is a draft written by the authors listed at the end of this document published by the Plattform Industrie 4.0 in November 2022.

Publisher

Plattform Industrie 4.0
Bülowstraße 78
10783 Berlin

Editing, layout and photos/graphics

Plattform Industrie 4.0
Bülowstraße 78
10783 Berlin

Status: October 6th, 2022

This document is published by Plattform Industrie 4.0. It is distributed free of charge and is not intended for sale.

Content

1 Introduction

Smart Manufacturing is a topical keyword, but what does it really mean? Most of all more adaptability and flexibility! Flexibility in terms of more product variants in one production line, smaller batch sizes of different variants produced at a reasonable price and reorganization of the production order with reduced effort. It also means, support during the engineering process to follow the fast changes of the market demands. To meet all those requirements, a machine-readable description of manufacturing functions[1] is required. This description has to be based on a well-defined model. The Capability, Skill and Service Model (CSS model) provides essential contributions to these challenges. This model combines the industrial requirements with initial practical experience, the current status of industrial research and the possible technological implementations.

This paper describes the CSS model. For this purpose, the motivation is first presented in Chapter 2 in connection with selected scenarios. This serves to characterize the CSS model related to industrial production. Chapter 3 offers at first a refined motivation and a general overview of the CSS model (Chapters 3.1 and 3.2) for a quick overview. The CSS model is explained in detail in Chapter 3.3 using a class diagram, starting from the product, process and resource model (PPR model) and then presenting the individual model components Capability, Skill and Service[2]. In addition to the structure of the CSS model, Chapter 3.4 describes activities for selected usage scenarios. Two examples, one from manufacturing and one from process engineering, illustrate how the model is embedded in these application domains. Chapter 4 describes a selection of concrete use cases that highlight potential usage scenarios. The CSS model can be implemented using various technologies. For this reason, essential technology references are made in Chapter 5. Chapter 6 gives an outlook for future work on the CSS model. Appendix A contains the definitions of the terms introduced in the paper in a more compact format to be used as a reference. The same applies to Appendix B, which contains the definitions of the activities.

[1] The term „function“ is used in a general sense in the paper.

[2] Here the term Service is a production service (see definition in annex); in German language it is the term “Dienstleistung”

2 Flexible Manufacturing in Industry 4.0

2.1 Motivation (Skills, Capabilities, Services)

An important trend in future manufacturing is the requirement for faster reaction to market uncertainties resulting in the need for more flexibility in industrial production. This flexibility concerns many different aspects e.g. the ability for fast introduction of new products or product variants, which affects the Product Lifecycle Management (PLM) process. Therefore, the possibility to efficiently produce high-mix scenarios under low-volume down to lot-size-1 is necessary, which requires new concepts for production control as well as the ability to react to problems and disturbances within a production and supply chains. All these challenges do not only require a new kind of cross-company collaboration, such as shared production using marketplaces, but also fast changeability of plants to efficiently adapt a production setup to changed conditions.

Today's production systems are limited to meet these requirements as typically there is a tight coupling of products and their required production process steps to the production resources.

There are no widely accepted or used standards applicable over all production sectors for the generalized description of products, production processes and functionalities of production resources, which enable decoupling and differentiation between these concepts and, as a result, a flexible relation between these elements.
This challenge has at least two dimensions. On the one hand, the distinction and relationship between product features, needed processes and process steps, provided and required functionalities, available interfaces etc. is often not clear or even not applied. On the other hand, there is a lack of concepts for a machine-interpretable description of these concepts which is restricting the ability to a dynamic and flexible matching between the required and offered functions and the derived automated decisions in a production control system. This allows the replacement of pre-programmed and pre-configured control sequences by model-based control design. The same is true for interfaces between production control and resources. In the case of changes in the use of a resource, no complex interventions in the control programs should be necessary. It should be possible to adjust the functionalities provided by the resource to changed requirements by configuration of the standardized interface.

To address these challenges and meet the requirements, fundamentally new concepts have to be introduced. Specifically, the decoupling of product design from production engineering is an important aspect. Introducing an abstract view on production functions is in line with the trend toward modularization of production resources, where the description of functionalities and standardized interfaces to encapsulated automation functions are key elements.

At the same time, a functional abstraction supports the trend from low level programming methods to modelling on a functional or even requirement level. This reduces engineering efforts in general, but specifically enables a fast reconfiguration of production systems without the need for extensive reprogramming.

In recent years, a large number of research activities took place to develop the required concepts in detail and elaborate their application in the industrial domain (Roman Froschauer, 2022).

Unfortunately, so far there is no common understanding with respect to naming conventions which allows a direct comparison of approaches and solutions and seamless cooperation between involved parties.

To improve this situation capability, skill and services are introduced.

The utilization within an overall general model is described within Chapter 3 of this document. But before scenarios give a context of the introduced CCS model. The definitions of these and other related elements are given in *Annex A – Terms* as reference.

2.2 Scenarios

Application scenarios as introduced by Plattform Industrie 4.0[3] (Plattform 2016) describe a vision for the digital transformation of industrial production as well as challenges connected with realization of these scenarios. Out of the scenarios described in (Plattform 2016), we will use the *order-driven production* and *adaptable factory* to explain which problems can be addressed with capabilities, skills and services.

Each of these scenarios shows a principal production context for the application of new conceptual approaches. The use case “Adaptable Factory” is related to the production process within one factory and the use case “Shared Production” is applied between factories.

Shared Production is an important part of the I4.0 defined “Order Driven Production” and focusses on flexible production configuration not only inside one factory but also including cross-site optimisation and cross-company collaboration to enable a fast reaction on changing markets which results in hard to predict long-term volume of incoming orders. It is stated that the core elements of this scenario are standardisation of both process steps of the product and self-descriptions of capabilities of production systems. The process step description references the product description (e.g. the electronic Bill of Material (eBoM)).

---

[3] https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/fortschreibung-anwendungsszenarien.pdf?__blob=publicationFile&v=7

These elements can obviously be mapped to the concepts of capabilities, skills and, in case of cross company setups, services. They enable the dynamic and possibly automated adaption of production processes. *Figure 1* shows the general applicability within the scenario.

The scenario **Adaptable Factory** is closely related to flexible production as well but focusses more on the production resources in an intra-company environment. The core element which supports flexibility is realized by a fast and efficient physical change of the production setup which usually is referred to as Plug-and-produce. This typically requires a modularization of production resources where skills and capabilities are important to ensure a well-defined self-description and interfaces on a functional abstraction level. *Figure 1* shows the interplay of both application scenarios and the utilization of skills and capabilities within this context.

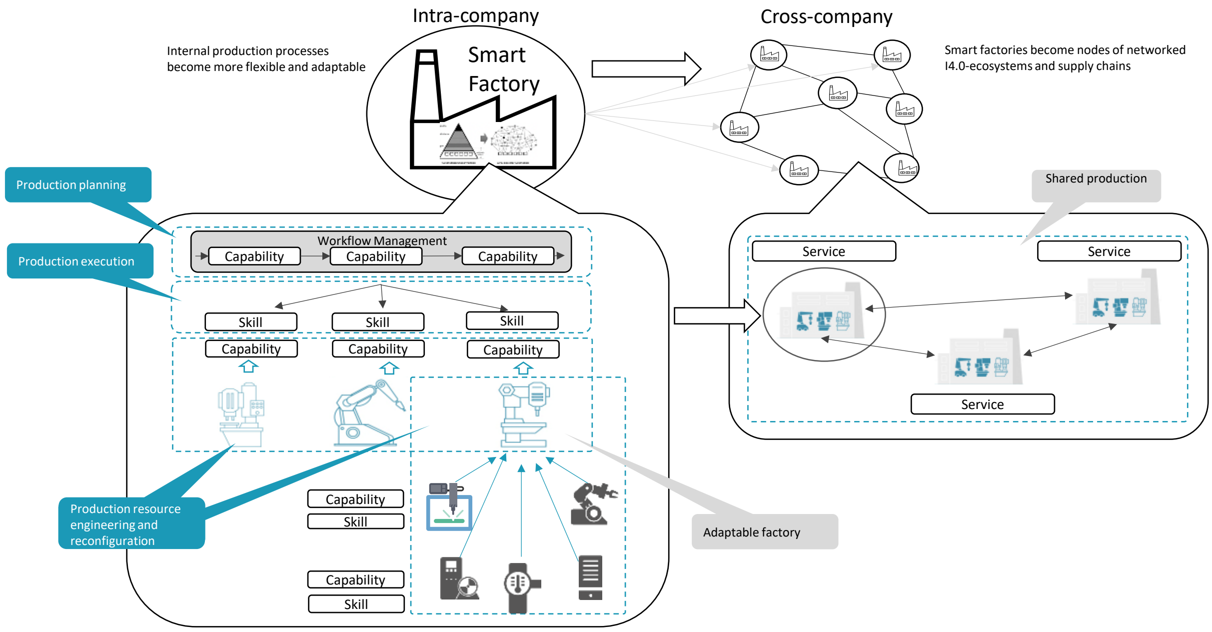


Figure 1: Capabilities, skills and services in the context of the Shared Production and Adaptable Factory scenarios [source: Chris Urban, Alexander Belyaev und Christian Diedrich. „Verwaltungsschale-basierter Ansatz für die Umsetzung von auftragsgesteuerter Produktion“. VDI-Kongress Automation 2022, Baden-Baden, Juni 2022]

The application scenarios describe a value network including business aspects and related challenges which contains several different process and lifecycle phases for products and production systems. Therefore, the deduction of specific technical challenges can be addressed by means of capabilities, skills and services. These scenarios have several aspects which at the same time connects both mentioned scenarios in case of production resource engineering and reconfiguration.

Scenario aspect A: Production planning
This scenario focuses on production planning starting from the products to be manufactured, i.e. product descriptions are matched on capabilities to generate a production plan). A process step description in terms of required capabilities may already exist. If planning takes place within one production site or a company, services play a minor role.
Challenges for the production planner: Due to various levels of planning, the level of capability abstraction and efficiency definition can vary due to conceptual, rough or detailed planning phase[4]. • Which capabilities are needed to produce the product? How can required capabilities be derived from the desired product features? What is the relation to processes and process steps? • How can the efficiency of the planning process be increased (based on capabilities)? • How can production processes be orchestrated with (offered) capabilities?
Scenario aspect B: Production resource engineering and reconfiguration
This scenario focuses on the development, engineering and commissioning of a production system. Using services, skills and capabilities describing elements of the system will help to execute these steps more efficiently. The operator of the production system will also benefit in case of reconfigurations of the system. This includes changes in an existing system to improve efficiency or react on (partial) failures as well as the expansion of a plant.
<u>Challenges for the production planner:</u> • Which processes are needed to ensure the desired product features? • Which resources with which functionalities are necessary for this? Are my resources able to meet these requirements? • What is the optimal plant for a process to be executed / a product to be manufactured? <u>Challenges for the plant design engineer</u> • How can necessary descriptions / models be made in parallel to the physical plant development? <u>Challenges for the plant operator (in case of reconfiguration):</u> • Which (existing) parts of the plant can replace failed parts? • Which plant components have to be newly procured?

[4] due to ISO 18828

Scenario aspect C: Production execution
This scenario focuses on production execution. The main goal is the efficient processing of the existing production orders (e.g. with regard to the overall make span) and an optimal usage of the production resources to achieve the respective KPIs (e.g. OEE, carbon foot-print, ...).
Challenges for the plant operator: • How can production processes be orchestrated with skills and executed efficiently? • How can the functionalities of machines be adapted or configured according to the current job without complicated modifications to the control programs? • How can the production process be optimized using existing degrees of freedom?

Scenario aspect D: Shared production
This scenario focuses on offering and requesting production orders via a marketplace. The objective of the scenario is to ensure the resilience of the production by enabling variable and in short-term changeable supply chains and making production resources accessible to production planers of other factories or plants. The factories and machines are seen as autonomous service units. A shared / standardized model of services and capabilities is required.
Challenges for the marketplace owner and participants: • What do descriptions of the offered / requested services / capabilities look like? • How is a comparison made between requested processes (special case: products) and offered processes?

Within the following Chapter 3, a model for capabilities, skills and services will be introduced as well as activities, where these artifacts will be used to address the challenges mentioned in the tables above. Within Chapter 4 we present generic use cases which can arise in the context of described scenarios and summarize the benefits resulting from an application of capabilities, skills and services.

3 Model for Capabilities, Skills and Services

In this chapter we present a conceptual model for capabilities, skills and services that provides the basic formal elements for the realization of scenarios of flexible manufacturing, as the ones described in the previous chapter. It is intended to serve the unification of terminology for the building of Industrie 4.0 applications with the goal of giving guidance to the naming and structuring of information represented in and exchanged between system components and stakeholders.

We motivate the necessity of formulating such a model in Section 3.1 and give an overview on basic design decisions in Section 3.2, before we lay out its details in Section 3.3.

3.1 Motivation

When looking at conceptual models that govern the representation of information in production-related applications we can observe that most approaches follow a common pattern of building on the notions of “product”, “process” and “resource”, which is often referred to as the *PPR paradigm*. These three notions form the conventional pillars for describing all kinds of scenarios related to production. The product pillar stands for all the materials and ingredients that go into the final product, ranging from a single screw or basic liquid as purchased from a catalog via intermediate stages of production like assemblies or mixtures all the way to the final product itself, such as an electronic component or a bottled beverage. The process pillar stands for all the activities that lead to the final product and drive production, breaking down the complete overall production process into sub-processes and finally into single operations, such as assembling two parts or heating a liquid. The Resource pillar stands for all the resources required to successfully conduct the production process, including the factory breaking down into production lines, work cells, stations, robots, tanks and so on, but also human workers who operate stations in semi-automated scenarios as well as supplementary materials like fuels etc. The PPR paradigm is thoroughly established in the literature and a description can e.g. be found in (Schleipen & Drath, 2009).

A typical pattern in conventional information modeling following the PPR paradigm is to tightly connect the process and resource dimensions[5]: for any operation in a process required for production, classical manufacturing planning assigns the machine most suitable for performing the operation as the resource to which the operation is to be allocated. Therefore, product design and manufacturing planning and execution are often tightly coupled in conventional planning. To realize a scenario of flexible manufacturing following the ideas of Industry 4.0, however, this tight coupling in the planning phase needs to be broken-up, so that the allocation of processes to resources can be decided on the fly and be automated to not require a human planner or at least give them automated support.

For this to happen, a common idea is to introduce an additional element in the middle between the process and its associated resources that mediates between the two and breaks their previously static linkage. Such an additional element on top of PPR has been called skill or capability or function and other names in the literature, for example in (Järvenpää, Siltala, & Lanz, 2016), (Pfrommer, Schleipen, & Beyerer, 2013), (ISO, 2004), (Solano, Romero, & Rosado, 2016), (Sarkar & Sormaz, 2019).

[5] For early phases of conceptual planning, tasks are sometimes allocated to categories of resources to be filled in by specific resources at a later stage. In such cases there is already an abstraction from rigid allocations that pro-vides some flexibility in current panning approaches. However, the capability concept goes farther than that.

All these different approaches have in common that they base the decoupling of processes from their associated production resources on a concept that captures the functionality to be addressed in the process, be it drilling, mixing, loading, or whatever other function the resource needs to fulfil. The different names introduced, such as skill, capability, functionality, etc. have caused some confusion and reflect the lack of terminological clarity in the literature. A more thorough discussion carried out within the ontology building community can be found in e.g. (Borgo, Terkaj, & Sanfilippo, 2021), where the notions of capability, capacity and functionality are discussed from a formal ontology point of view.

To meet this confusion, our intention here is to streamline the terminology in these approaches and to define a vocabulary that unifies their usage. The resulting conceptual model is an extension of the well-established PPR representation paradigm extended by the notions of “capability”, “skill” and “service”, all of which capture the underlying production-relevant function at a different perspective. Therefore, we call the resulting model the Capability-Skill-Service-Model, or CSS Model for short.

3.2 Model Overview

The CSS Model covers four areas (each area is visualised with a separate color in Figure 2). First, the yellow boxes describe the PPR model. Capabilities, in the orange boxes, describe functions reflected in production process steps, which on the one hand are required in order to create products and on the other hand are carried out on operating resources. Required and provided capabilities must be matched in order to find candidates for a suitable sequence of production steps for given requirements. This can initially be done on a descriptive level – e.g. by comparing capability types and their properties – regardless of which actual resources execute these process steps later.

In order to apply the function described by a capability in a specific production process step, a so-called "Skill" is invoked, which is an implementation of this function provided by a specific resource (blue in Figure 2). A skill is activated and controlled through an interface provided by the skill implementation in order to allow external entities accessing a skill without giving away internal implementation details. Skills are assigned to resource instances.

In addition to the functional aspects described by capabilities, other organizational and commercial aspects, such as timing, quality or cost must also be considered. For this purpose, required and offered services (green in Figure 2) are defined in the model. Services allow for the offering of (sets of) capabilities in a broader scope of larger supply-chain networks that go beyond a single capability's local production setup. Thus, services are usually not part of a resource, but of higher-order software components such as Enterprise Resource Planning (ERP) systems.

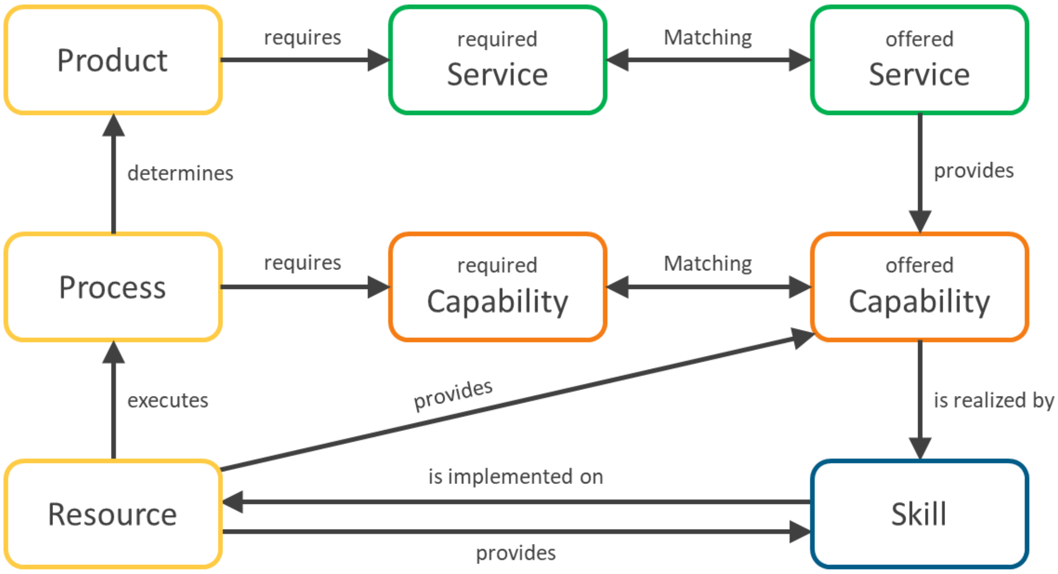


Figure 2: Simplified overview on most important aspects of the CSS model.

3.3 Model Details

In this section, we present a conceptual model for skills, capabilities and services as the main contribution of this paper, which details the previous simplified scheme in *Figure 2* into a more formalized specification. The model adds the notion of "function", understood in the context of production, to the classical PPR paradigm and differentiates it into the three different levels of technical implementation / invocation (skill), abstract description (capability) and bundled market offering (service). Although we present this conceptual model in form of a UML class diagram, it is not meant to be the technical specification of a schema for immediate implementation in a programming language or data management framework. Rather, it is supposed to clarify terminology of the things relevant for production scenarios. Our design goal is to take a minimalistic view and include only those terms that are required for such scenarios and present them in their most general way. In this sense, the model identifies the necessary basic elements that require instantiation for realizing flexible production and the relations that hold between them – it does not further detail these basic elements by means of specialization. Such detailing is left for more specific models derived from this one for the sake of capturing specific industries, industrial domains or even concrete use cases and application scenarios.

Figure 3 depicts the conceptual model where nodes represent instantiable entities (UML classes) and labelled arrows represent relations (named, directed UML associations) between these entities with the name clarifying the nature of the relationship when read in the direction of the arrow. Any future model that is to be implemented in a data management framework of a database, knowledge graph or API will resemble this UML specification, but will need to add more technical detail depending on the use case at hand.

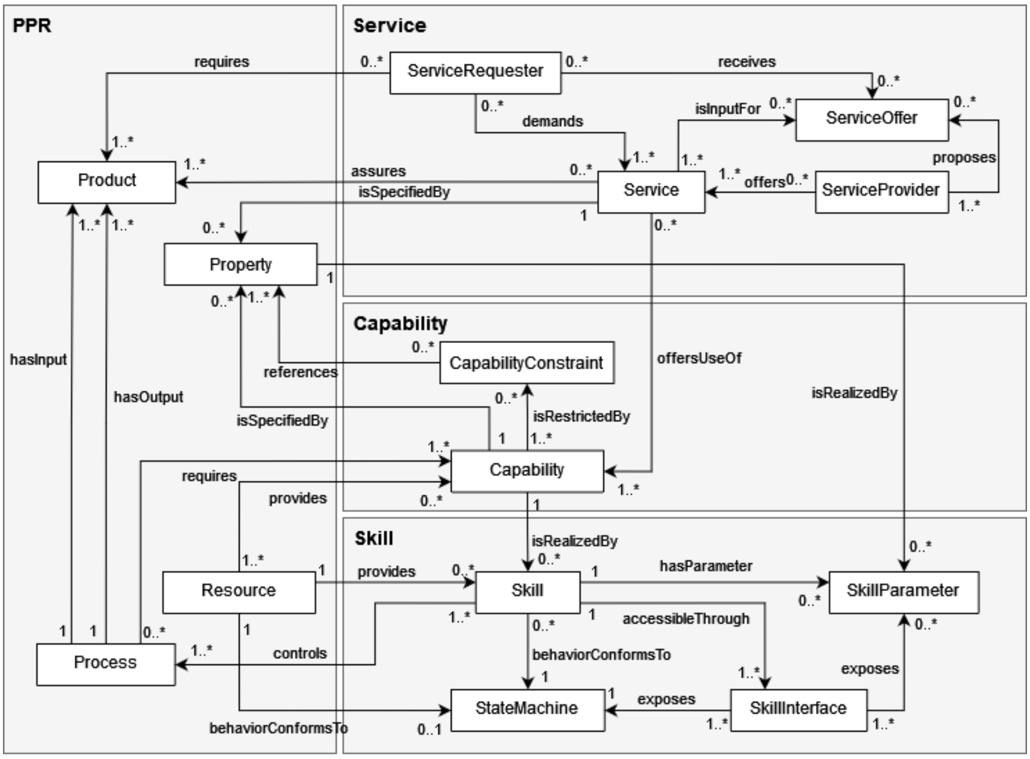


Figure 3: CSS-Model showing relevant terms and their relations as well as references to the basic aspects of PPR.

In the following subsections, the four model aspects of PPR, Service, Capability and Skill are described in more detail, while the complete textual definitions of all terms can be found in the appendix.

3.3.1 Product, Process and Resource

Building on the prominent Product-Process-Resource (PPR) representation paradigm described in [Schleipen & Drath, 2009], we include its name-giving concepts into our model to capture Product, Process, and Resource in their most generic form (see 3.3.1). Properties are common model elements which specify the capabilities and services and are realized by the skill parameter. But once the CSS model is being further specialized for certain use cases, then more specific subclasses can be introduced for sub-processes, different types of product parts, such as assemblies or raw materials, or specific types of resources like transportation systems, manufacturing stations or tools. The notion of a “function”, which is used throughout this document to define capabilities and skills, can also be illustrated by PPR. A function can be seen as some kind of abstract process step that is not yet tied to a specific resource, but has a description of its inputs and achievable effects. These effects typically have an impact on the processing state of products.

3.3.2 Capability

Capabilities are production-relevant abstractions of functions applied in the context of a process step. While capabilities will typically specify production functions with an effect in the physical or virtual world, software functions that only apply to the virtual world may also be modelled as capabilities. Capabilities that specify a production function reference terms of an actual manufacturing method, such as “drilling”, together with properties and constraints on them for detailing the restrictions of application. An example of a capability could read “drilling a hole with a depth of max. 20cm, diameter of max. 10mm and with tolerance +/- 0.1 mm into certain types of metals”. Capabilities are provided by production resources that claim the ability to apply the expressed function. On the other hand, they are required by a process as part of a product design’s functional requirements. In these two roles, capabilities are utilized for testing whether the functions provided by production resources match with those required for a production request.

Furthermore, capabilities are implemented by means of skills, which contain details at the level of implementation and invocation of automation functions. Additionally, capabilities are offered in terms of services to stakeholders in a broader supply chain network outside an production setup with one shopfloor.

3.3.3 Skill

A Skill is an implementation of a function specified through a capability that is deployed on a specific resource[6]. Skills encapsulate the internal complexity of functions and provide an interface to be invoked by other systems such as MES and superordinate controllers.

Skills typically have parameters, which are either input parameters or results, allowing to command or monitor the execution of the skill. Parameters can influence different aspects of the production for example the process directly and the control execution. Parameters

[6] Notice that skills are described from a rather static point of view not considering the dynamics of their enablement and availability in terms of setup operations like changing the tool of a machine. Although not described in detail here, this aspect is to be positioned within the activities that deal with the static model elements and is very important for handling skills in practice.

may correspond to resource properties, i.e. a skill parameter can be the representation of a capability property in a particular implementation technology which allows to set or get values of a capability property. The behavior of a skill corresponds to a standardized state machine, so that the current state of a skill as well as possible interactions are transparent to systems interacting with the skill.

Every skill shall be accessible through at least one interface, which are network or programming access points between a skill and other systems. A skill interface needs to expose ways of interacting with a skill's state machine as well as its parameters. One possible implementation of a skill interface is an OPC UA server which exposes OPC UA variables for all skill parameters as well as methods to invoke transitions on the skill's state machine. Another one is the "Control Component" which is a result of the Basys 4.2 project []. This is currently under Submodel-Template development at IDTA.

Distinguishing between capabilities and skills ensures that a function can be implemented using different programming languages and technologies, depending on the target platform. The additional separation between a skill implementation and a skill interface also allows a skill to be exposed via different interaction mechanisms (e.g. PROFINET, OPC UA or as a web service).

3.3.4 Service

A service specifies the means of provision of one or more capabilities offered by a service provider to a service requester and extends its description with commercial aspects. It is used in particular to offer capabilities across shopfloor boundaries, for example on a manufacturing marketplace. The term service in the CSS model is thus more closely aligned with the service concept of economics and clearly differs from the service concept of information technology (see distinction in Section 5).

There are application scenarios, which involve some kind of automated order processing in supply chains spanning across various companies (see Section 2.2). For these scenarios, additional characteristics and aspects besides capabilities and their constraints are important. These so-called service properties typically go beyond the description of purely technical capabilities and may include, for example, economic criteria such as delivery dates, cost and agreements regarding documentation or maintenance.

Services are demanded by a service requester who provides a specification of a requested service including service properties. If a service provider can provide the demanded service, it can propose an offer as the basis of a binding contract to execute one or more services. The service requester can then accept the proposal in a specified time period and under the proposed conditions. If service requesters are searching for services through a marketplace, multiple service offers may be created by different service providers. These offers could be mutually exclusive, or they could also be combined together to fulfil a requested service.

3.4 Activities

Based on the CSS model, this section describes how the concepts of services, capabilities and skills are utilized for the realization of production scenarios. For this purpose, we present activities to which elements from the CSS model serve as either input or output, making clear what skills, capabilities and services are being used for in which phase of interaction. The focus is on the planning phase and describes activities ranging from formulating the service demand to the verification of feasibility of a specific manufacturing process with a defined set of skills.

All activities are presented as a UML activity diagram (*Figure 4*). Depending on the specific use-case, it might be sufficient to only use a subset of the defined activities. Each activity can be used independently of the general flow shown in the activity diagram. To ensure readability the overview of the activities in Figure 4 only shows the successful selection of services, capabilities and skills. The Shared Production scenario serves as basis for the description of activities, as it shows a broad coverage of functionalities and is a good representative for the scenarios presented in Chapter 2.2. This scenario is intended as an example for a better understanding of the CSS model.

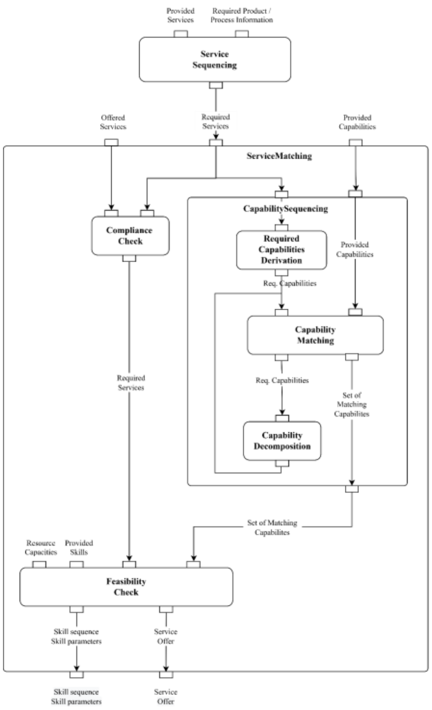


Figure 4: Activity diagram representing the most important activities related to elements of the CSS model.

Companies searching for possible manufacturers to produce a product act as a ServiceRequester. Trusted partners (acting as ServiceProviders) share manufacturing services and free capacities via a shared platform or through digital interfaces. A ServiceRequester may not be able to precisely describe requested services. Instead, a ServiceRequester may typically request a set of required processes or a product. This product may be described with certain product properties or also on a more generic level, e.g., referencing only certain geometric features. In addition to that information on product or process level, resepctively, a ServiceRequester may also define other service properties such as the commercial conditions (dates, pricing, quality).

The activity ServiceSequencing is needed in order to select one or more required services for the rather abstract service description given by a ServiceRequester. Through ServiceSequencing, a sequence which ensures manufacturing of the product as specified by the ServiceRequester may be derived. Derived Services can be defined on different levels of abstraction. For example, the derived ServiceSequencing for producing a metal piece could first require the service of CNC-milling or, on a finer granularity, a set of milling services for single features such as pockets or holes and subsequently the service of a special coating (as finishing).

The composed activity ServiceMatching is responsible for finding ServiceProviders that are able to provide the services derived during ServiceSequencing. For each service, ServiceMatching compares required to offered services with the objective to create a binding offer. While ServiceMatching depends on all its sub-activities such as CapabilitySequencing, those sub-activities can be used without ServiceMatching (e.g., in cases of internal production planning without service aspects). To ensure an efficient and correct Service Matching, ServiceProperties should follow an agreed semantics.

ServiceProperties comprise non-product and non-process related aspects such as compliance with supply chain law, country embargos or environmental criteria. Based on the ServiceProperties, the ComplianceCheck verifies the adherence of the ServiceProvider to those defined set of rules or laws. ComplianceChecks can evaluate aspects such as fulfillment of the European Supply Chain Act, the Climate Change Score of the Carbon Disclosure Project, quality characteristics or certification according to the Responsible Minerals Initiative (RMI).

Based on the product requirements defined in the required service, one or more capabilities are needed. The activity CapabilitySequencing derives the required capabilities and their sequence and selects suitable offered capabilities.

For example, a service CNC-milling may be decomposed into several milling, drilling and grinding capabilities which can be performed in different sequences. CapabilitySequencing[7] combines multiple sub-activities. For large companies, the solution space can grow extremely large, and hence CapabilitySequencing may include optional optimization steps. There are also cases where service requirements can only be partially fulfilled, for example because a complex surface treatment is required. In this case CapabilitySequencing may also result in an incomplete fulfillment of the product requirements so that product requirements must be altered, or additional ServiceProviders must be found.

To ensure compatibility of different service and capability descriptions, the activity RequiredCapabilityDerivation derives the necessary capabilities to achieve a desired effect required by one or more products in a particular state to transition to one or more products in a subsequent state. For example, if the product requirements define the features of a product, it may be necessary to derive the capabilities described as production processes. The feature hole could be mapped for example to processes such as milling, drilling or laser cutting. This activity may be omitted if the required product requirements and capabilities are already described based on compatible concepts.

Based on required and offered capabilities, the activity CapabilityMatching identifies resources that fulfill required capabilities. This involves ensuring that a given set of CapabilityConstraints is fulfilled. The activity may be done iteratively with adjustment of the required CapabilityConstraints depending on the result of the CapabilityMatching until a set of matched capabilities is derived. Because the granularity of capabilities is not defined, the resources to be identified may be individual assets or groups of such assets, forming for example a supply chain.

As required and offered capabilities may be described at different levels of granularity, it may not be possible to match the two concepts directly. The activity CapabilityDecomposition breaks a capability down into subordinate or part capabilities and thus enables deriving a set of new capabilities which are composed of one or more other capabilities. An example from assembly is a pick & place capability which is composed of linear movement of an axis and linear movement of a gripper. CapabilityDecomposition also defines the order of the subordinate or part capabilities either explicitly or implicitly using CapabilityConstraints. The resulting required set of capabilities is input for CapabilityMatching.

[7] Sequencing includes general ordering and defining predecessors and successors.

If a matching capability sequence is derived by the upstream activities, a FeasibilityCheck assesses the possibility to achieve the desired effect with a skill execution in a concrete production context. Based on a matching capability sequence, service properties and resource capacities, this activity assures or validates the concrete execution of a skill. That could include, for example, trajectory planning or selecting valid process parameters. Concerning for example milling of a pocket, the FeasibilityCheck might derive the specific cutting parameters and strategy. It might also check the effect of the skill execution such as assuring that there are no thin walls which can lead to product defects after performing the milling process. The derived process parameters can be used to calculate process times and costs of the skill execution. By comparing the resource capacities and required process times, the skill execution can be assured and an offer can be calculated. There might be one FeasibilityCheck for all the skills of the previously found capability sequence or multiple FeasibilityChecks for each corresponding skill in that sequence. The result of the activity FeasibilityCheck is a sequence of validated skills with parameters and an offer to perform the specific sequence of skills.

Based on the presented activities, other activities of production planning and control may need to be performed. A skill sequence and the corresponding parameters would also allow for automated scheduling and execution of the skills via the SkillInterface. However, due to the focus on planning processes, those activities are not described further.

4 Usage of Skills/Capabilities/Services

In this chapter we present some typical generic use cases, which can benefit from the usage of capabilities and skills as well as the corresponding services in shared production scenarios. We first list these use cases in a systematic form in Section 4.1 and then describe elaborate usage examples from the two different industries of discrete manufacturing and process industries in Section 4.2.

4.1 Generic Use Cases

The following table contains a set of generic use cases that can arise in the context of scenarios as the ones described in Chapter 2. Each use case concerns a situation in which the notions of skill, capability and service are being applied by certain users in specific roles in order to gain a benefit in flexible production.

Every use case features an expressive title as well as a description. Additionally, the involved artifacts of skill, capability and service are shown and relations to the activities described in Section 3.4 are highlighted. Use cases are described in the style of a typical user story to capture user role, the user's intention and expected system behavior. Moreover, the use cases are grouped into categories that have a different focus on automated production planning, resource management and production optimization and execution.

ID	Title	Use Case	Artefact usage including activities
Group 1 (Semi-) Automated Production Planning			
UC1.1	Assess principle producibility	As a production planner I want to check whether a certain product (BOM/BOP/recipe – bill of process (BoP) can be produced on a specific target production facility in general, and if not, I want to know what the reason for a non- producibility. This result can either support the further offering/planning process ("offline" case) or used as an input of the detailed in-house production planning/control ("online" case).	Required capabilities are matched against offered capabilities to verify their compatibility (CapabilityMatching). If the required capabilities are described on a different abstraction level as the offered capabilities a CapabilityDecomposition is required.
UC1.2	Find production services via market place	As a product owner/product manager I want to support an outsourcing decision of manufacturing steps or a whole product and identify suitable other manufacturers or supply chains (that have the right machine tools, capacities, etc.) e.g., via a marketplace based on the manufacturing requirements of my product design in order to complement my portfolio or to be more efficient. As a production planner I want to support this process in defining requirements for the production described as capabilities. As a production owner I want to offer manufacturing steps to other manufactures (production or machine as a service).	The requested Product Properties and their commercial conditions are matched with the offered services and a sequence is derived (ServiceSequencing). Based on the sequence the required and offered Services are matched (ServiceMatching). The ServiceMatching is composed of several sub-activities and consists of some of the other use-cases such as UC1.1 and 1.3.
UC1.3	Generate eBOP	As a production planner, I want to generate potential electronic BoP (eBOPs - recipes) (independent of the available resources) for a certain product based on an available product design (BOM) in order to automate the early production planning process.	Sequences of required capabilities are derived from product design information such as geometry in CAD, etc. (RequiredCapabilityDerivation).
UC1.4	Generate management BoP (mBOP)	As a production planner, I want to generate potential production processes (routes through the plant) for a given product/order taking into account an available product design (BOM) and the available resources (with skills) and possibly the current capacities (online case), in order to automate the concrete production planning process (including the allocation of process steps to resources).	The use-case represents an extension of UC1.3. Based on the required Capabilities and the offered capabilities a set of matching capabilities is derived. If the required capabilities are described on a different abstraction level as the offered capabilities a CapabilityDecomposition might be needed. To take specific resources into consideration, the FeasibilityCheck might be necessary to confirm the specific technical and organizational feasibility to use the resource as part of the potential route through the plant.

Group 2 Resource Management and Production Configuration and Commissioning			
UC2.1	Check (Re-) Configuration of Production Design	As a production planner, I want to check a given production setup (or alternatives of it) based on a mBoP with respect to producibility of a given product (variant) mix and resulting KPIs determining the efficiency of the production. Information from component catalogues can provide additional production options with higher efficiencies.	The use-case represents an extension of UC1.3. The CapabilityMatching which is used to determine the producibility in the context of a production setup results in a set of matching capabilities. Via the FeasibilityCheck specific KPIs are calculated which can be used to determine the efficiency of production.
UC2.2	Plug and Produce	As a production operator, I want to change my production facility setup by easily combining, adding and removing production resources in order to build up flexible production workflows or exchange production units with minimal (ideally zero) effort.	Capabilities and skills provided with the machine to be added to a plant are recognized and made available for production execution. Using the results of the FeasibilityCheck factory/production line layouts can be evaluated.
UC2.3	Production Commissioning	As a production engineer, I want to setup and test a production system consisting of resources/machines which provide skills to be orchestrated by the production control system in order to test whether my planned setup runs on my real production facility or (in the case of virtual commissioning) on a simulated environment.	Derive skill parameters using the FeasibilityCheck and test the invocation of skill interfaces required for a given production scenario. The invocation can be performed in a simulated environment for testing and debugging purposes.
UC2.4	Ressource solution finding “Catalog”	As a machine building engineer in development phase of a new machine or as production planner in planning phase of a new product or as machine operator, I want to check the availability of equipment (automation products / resources) for a specific process/process step in my application on capability/skill basis with the help of digital and web-based services (catalog services, sizing services, solution finding services) of machine equipment suppliers. This check can be requested manually or automatically by engineering tools or even by a machine administration software.	Required capabilities are matched against offered capabilities to verify their compatibility Additionally, required attributes are matched against offered attributes.
Group 3 Production Monitoring and Execution Optimization			
UC3.1	React to Disturbances	As a production operator, I want the production system to identify disturbances autonomously (possibly involving user interaction for critical decisions) and overcome them by generation and execution of alternative production processes (paths) in order to keep the production running.	In case of disturbance a replanning is necessary. Based on the existing capability sequence of the disrupted production path, a FeasibilityCheck is performed with the goal to identify alternative skill sequences and the corresponding skill parameters. If an alternative is found, the production workflows and calls to skill interfaces are being rerouted for their automated execution.
UC3.2	Monitor Production	As a plant operator, I want to monitor the production process (based on the execution of skills of a set of resources) and to visualize the results on	The SkillParameters derived by FeasibilityCheck (OutputParameters) and skill states involved in a running production scenario are being traced to form a basis for monitoring more abstract than low-level device signals.

		any system (e.g., mobile device, MOM – Manufacturing Operation Management/MES) to track usage, detect failures or to predict maintenance.	
UC3.3	Optimize Production Workflow	As a production planner/operator I want to optimize my production workflow (for a product) in terms of QoS criteria, like throughput, energy consumption, either as part of planning (offline case) or dynamically during operations (online case). It has to be noted that the production planner can provide input and follow given KPIs, but as these optimizations may affect many departments of a company this has to be done in coordination with other persons responsible for storage, logistics, energy etc.	Capabilities are being matched and associated and skill invocations are being checked for testing validity of alternative production workflows. Based on the FeasibilityCheck, a set of skill sequences and their corresponding KPIs are derived. The different skill sequences are then compared based on different objective functions.
UC3.4	Schedule production execution	As a production planner, I want to generate a principle/possible production schedule for a given product/order stack in my production facilities in order to optimize automated production.	Based on CapabilityMatching and FeasibilityCheck, a set of skill sequences and their corresponding KPIs are derived for all products which should be produced in a certain time period. This information is used to perform the scheduling.
UC3.5	Utilize after sales services	As a manufacturer, I want to use After-sales services e.g. maintenance-repair-overhaul activities or value stream management which are not directly related to production in order to identify optimization potential in my production (e.g. to improve the production/material/information flow).	After-sales services or value stream need a combination of skills which are realized within the real production or in corresponding digital twins to gather the required information. Based on the FeasibilityCheck a set of skill sequences is derived. This information is used to support e.g. a value stream analysis. Relation to: UC 2.1, UC 2.3, UC 3.2, UC 3.3

Table 1: Generic use cases for capabilities, skills and services

The use cases in Group 1 "Automated Production Planning" are all concerned with the early planning phase, where users are being supported in taking production planning decisions or generating planning artefacts. The user role typically addressed here is that of a production planner whose task is to plan and coordinate mainly in-house production. For this planning phase, the primary interesting elements from the CSS model are the Capability for matching and sequencing and the Service for outsourcing decisions. The production planer can benefit from the application of capabilities by having a formalized and (possibly) standardized way to relate requirements for production and possibilities of production systems thus shorten the time for introduction of new products of product variants. In case of outsourcing of production steps services (including capability descriptions) are a prerequisite to enable efficient communication between request and offering side (e.g. via market places).

The use cases in Group 2 “Resource Management and Production Configuration and Commissioning” are focused on the set-up of in-house production plants and their proper functioning. Here, typically a production operator is supported in finding the right setup of a plant or its suitable configuration for incoming production requests – instead of planning artefacts the target here is the setup of production resources itself and their changeability aspects. Besides capabilities, also skills play a role at this level, as the proper functioning of resources needs to be guaranteed. Having a common approach to semantically describe the function of resources with offered capabilities as well as their invocation with skill interfaces will ease engineering and supports modularization approaches.

The use cases in Group 3 “Production Monitoring and Execution Optimization” are primarily focused on the running processes in the operations phase of production plants, but can also extend to complete supply chain networks. Here, a mix of skills capabilities and services can be utilized to support a broad range of user roles and ensure that production runs properly and meets certain requirements and KPIs.

4.2 Application examples

Within this section we will provide two application examples, one from manufacturing and one from process engineering. It is intended that understanding the application of model elements within these examples should help to better understand the concept of the capability model. Furthermore, the application examples serve as a concrete instantiation of selected generic use cases presented in Chapter 4.2 (mainly focusing on production planning).

4.2.1 Manufacturing Example

The capability-based description of products can reduce the required expert effort in the planning phase of manufacturing processes significantly. In a shared production scenario, a customer designs a part and searches automatically for a possible manufacturing strategy (UC1.2). The designed part is automatically analyzed and required technologies for manufacturing are identified. Companies (service providers) offer certain manufacturing technologies as a service through a network (e.g. Gaia-X/CATENA X). In this example, a shared production network is described where an individual small truck[8] is produced.

A service provider is identified in the production network for each component to be produced. One possible configuration of the truck contains a trailer that needs to be manufactured by machining. The trailer can be individualized by a customer and simulates a part that is produced in lot size 1.

[8] Die Shared Production Kaiserslautern - SmartFactory-KL

When the production order is placed, the network is searched for a service provider that offers the machining of the trailer. The service provider receives the product description of the truck. The product description contains information about the required material, the outer contour, and the geometric features (drilling holes, pockets, slots, etc.) of the product (*Figure 5*). The geometric features are mapped to the capabilities of the production environment. (UC1.1/1.3) The naming of these geometric features, capabilities, and skills was kept identical to simplify the matching at the beginning. The capabilities of the production environment are described in an Asset Administration Shell (AAS) and refer to a skill. The skill is implemented on a milling machine and is accessible through an OPC UA skill interface. In the future, ontologies will be integrated into the system to link different names and standards of geometric features and capabilities. The capability “hole drilling” can be limited by properties such as a range of possible diameters, a maximum depth or process tolerances. Capability Properties also describe the needed input Parameters of a skill. Several parameters are necessary to start a skill. For example, for the skill parameters of a drilling hole, the position, orientation, diameter, depth, and quality requirements are needed. Before the borehole skill can be executed, a feasibility check is performed (UC1.4) (Volkmann, 2021). Here, the parameters of the skill are transferred to the machine. The machine checks whether production with the desired parameters is possible. If the check is successful, the required duration, costs, and energy consumption of the machine are determined. The feasibility check is in this case needed because every product is individual. After successful execution of the feasibility check, a skill can be started by OPC UA method calls. The machine, a representative software entity or a human plans the toolpaths and required tools independently. This allows the identification of individual processes for individual products. The implementation is described in an online video[9].

[9] SmartFactory-KL-Live Skill-based Production: Skill-based Production in der industriellen Anwendung. Ein autonomer Roboter bei der Arbeit. – YouTube (https://www.youtube.com/watch?v=P3KsxH3eLng)

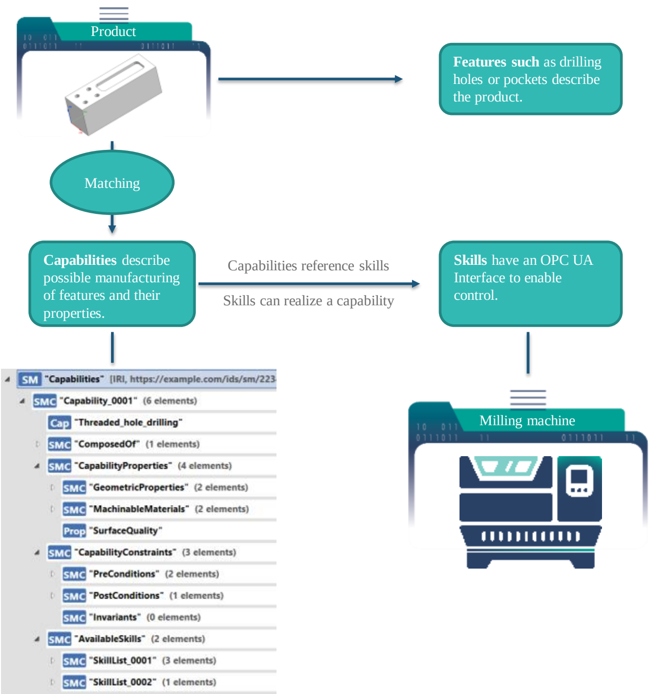


Figure 5: Overview of the manufacturing example

4.2.2 Process engineering example

This example description is intended to provide a reference from the CSS model to process engineering. Thereby a modular principle is used due to desired modularization of plants. The construction of modules intends to reduce the planning effort required for combining suitable modules. The modules are pillars of a hexagonal base area. Each module fulfills at least one process-based operation and is able to operate autonomously as well as hold its own automation software and structural information. The continuous use of a service-oriented architecture ensures a high level of flexibility with respect to different software structures. This allows the implementation and validation of diverse automation concepts for modular plants[10].

One of the modules is the Honeycomb 20, see Figure 6 right. The Honeycomb 20 consists of the components or resources: stirrer, pump, five valves, level sensor and flow sensor, see Figure 6 left. The combination and aggregation of components enables modular process units, known as Process Equipment Assembly (PEA).

[10] https://www.plt.rwth-aachen.de/cms/plt/Forschung/Anlagen/~hoda/M4P-AC/?lidx=1

A PEA is developed once and contains the physical design of the process step to be implemented as well as the information technology interface to higher-level systems. The description of a modular process units is provided with the module type package (MTP). MTP defines and describes data of the structure, an information interfaces, process sequences and functions (MTP services in terms of the MTP specification, not in terms of the CSS model described in Chapter 3, see explanation in the next paragraph) of modules from automation technology. [VDI/VDE/NAMUR 2658-1] In the context of the CSS model, the implemented MTP service with its procedure is to be defined as a skill.

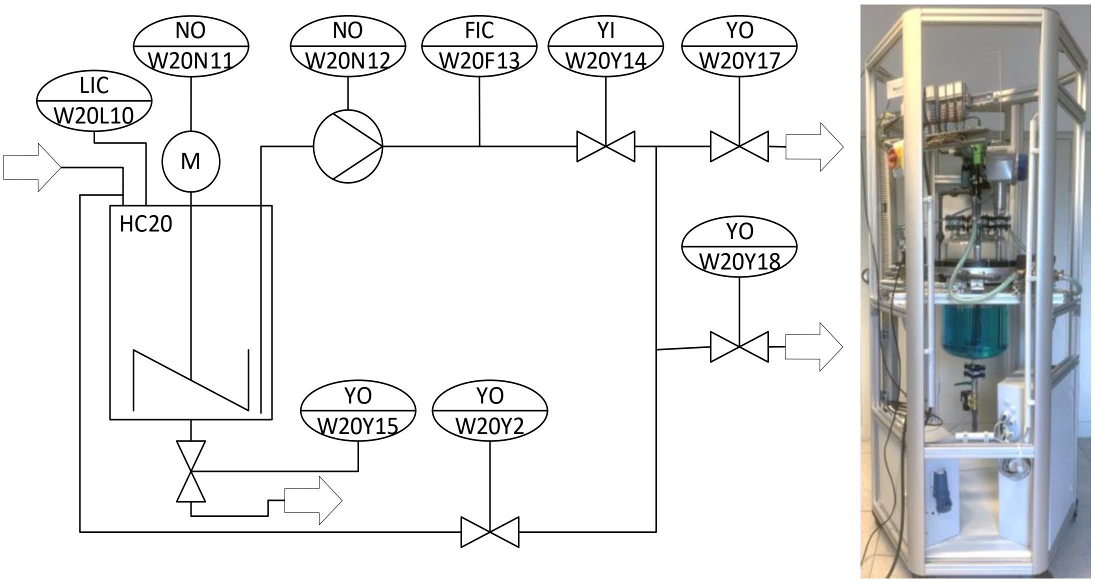


Figure 6: Module of a process control plant named Honeycomb 20 (left: P&ID, right: Photo)

A description of the module is contained in the MTP due to the use of module integration into a modular process plant. Modules provide MTP services with a predefined behavior (procedure) as well as a standardized interface, that are offered externally with their description as MTP. These Modules can be accessed or executed via a higher-level system. ([VDI/VDE/NAMUR 2658-1])

The MTP service of a module can be related to the beforementioned offered capability that was described in Chapter 3. A description of an MTP service can be considered as a capability. A Skill is the executable implementation of an encapsulated (automation) function specified by a capability of a PEA, which is the implemented MTP service with its procedure. A PEA is the resource of the CSS model which can provide skills in form of MTP services with its procedures and their capabilities.

Further, higher level capabilities can be generated with the aggregation of additional modules into the process plant. If the capsulated module provides implemented process engineering functionalities as MTP services (skills) to a superordinate unit, then it assumes the role of a skill provider. The MTP service offered by the module can be accessed by the unit and is thus a skill user. [VDI/VDE/NAMUR 2658-1] In contrast to the service of the CSS model, the MTP service is at skill level and does not consider business, compliance or any commercial aspects.

The development of a process can be conducted based on the standard ANSI/ISA–88. This standard provides information about production requirements for a specific product or intermediate product, which are known as a recipe. The information of the recipes describes the sequence of the process to produce the process product on different levels. The general recipe is applicable at company level and is used as a basis for subordinate recipes. The general recipe is created without specific knowledge or information about the PEA used to produce a product. (UC1.3) It identifies raw materials, their relative quantities, and the processing required, but without specific reference to a particular site or the available resources. [ANSI/ISA-88]
The required processing in the general recipe can be related to the required capabilities of the CSS model. The use of standards with its functional terms enables the assignment to the corresponding capabilities. Thus, the required capabilities from a process description (e.g. a general recipe) can be matched with the assured capabilities of the respective resource. (UC1.1)

Due to the consideration of other organizational and commercial boundary conditions a site recipe is necessary. The site recipe is specific to a particular site, but not specific to a particular set of process cell equipment. Thus, the site recipe is the combination of site-specific and general recipe information. It can therefore be derived from a general recipe to meet the conditions at a specific production site. [ANSI/ISA-88] With this recipe, aspects of service requirements from the CSS model can be met, which can be matched to the offered services. (UC1.1)

A master recipe is specific to the equipment (resource at the CSS model and PEA at the MTP description), raw materials and capabilities of a process cell or a subset of the equipment of a process cell. For example, at the master recipe level, it is important that the recipe match the capabilities of the equipment to the required capabilities from the general recipe. A master recipe can be derived from general or site-specific recipe information [ANSI/ISA-88]. This recipe provides assured capabilities of their resources which are realized by skills. (UC1.4)

The process engineering functions provided in the module of this example are encapsulated as MTP-services, e.g., reactor module with a mixer offers the MTP-service "Mixing", filling reactants into the reactor offers the MTP-service “Filling”. Furthermore, an MTP-service behaves with the state machine from VDI/VDE/NAMUR 2658-4.

5 Related Approaches and Technologies

In this chapter, we position the previously described CSS Model in the light of prominent technologies and conceptual approaches that are currently being discussed for the realization of Industry 4.0 systems. For each of these we indicate their relevance and possible role they can play with respect to implementing capabilities, skills and services in industrial systems.

5.1 Asset Administration Shell

The Asset Administration Shell (AAS) is a core concept of Industrie 4.0 to manage the interactions with Digital Twins (DTs), as proposed by the Plattform Industrie 4.0 consortium[11]. It provides a single-entry point to the information connected to the DT of an asset. This information is captured in so-called submodels, which represent different aspects of the information of an asset. Every submodel is composed of submodel elements where each element exposes its meaning by referencing a concept repository. A submodel typically should be based on a submodel template.

In this sense, the capability part of the CSS Model as described in Section 3.3 can serve as a basic schema for information models maintained in the concept repositories that are used in AAS submodel templates. Specific submodels can then describe DT content in terms of capabilities and their relation to surrounding elements such as those from PPR.

The CSS Model can build the basis for capability related AAS submodel templates. The elements in these submodels can reference to elements of repositories. Development of an explicit submodel template for “capabilities” is currently ongoing in a sub-group within the IDTA organization[12], with participation of some of the authors of this paper.

BaSys 4.0 Control Components expose, in a unified manner, offered capabilities of single or group control units in the device layer of BaSys-4.0 compliant production systems. BaSys 4.0 Control Components meet specific requirements where each component instance presents itself as a BaSys-4.0 service system participant that complies with the standards associated with this role.

The BaSys 4.0 Control Component is a means to basically represent, implement and offer possi-ble actuations of a plant in form of capabilities. The components connect to the controlled process via IO interfaces and, as an element in a control hierarchy, to subordinate or superordinate con-trol units, respectively, via respective “service interfaces” to provide or

[11] https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html

[12] Submodels - IDTA English (industrialdigitaltwin.org)

call offered capabilities. The various capabilities of an individual component are implemented by operations (skills) that are callable via the components "service interface". The invocation of an operation is referred to as an ORDER in BaSys 4.0.

The components can, e.g., be programmed using the standardized PLC programming paradigms of IEC61131-3 for cyclic execution and IEC61499 for event-based execution. Examples can, e.g., be basic skills like "go-to-position" or "grip" of a mobile gripper unit, however, can also be significantly more (e.g., the capability "assemble" of an advanced Pick&Place station) or less complex. These capabilities are advertised via the "service interface", e.g., using OPC UA, to superordinate Control Components and to the Middleware. Superordinate Control Components combine different subordinate components into composed capabilities.

5.2 Semantic Web Technologies

Semantic Web Technologies provide mechanisms for knowledge representation in information systems based on a stack of downward-compatible languages for information models and knowledge representation standardized by the W3C[13]. Ontologies constitute reusable information models that capture the knowledge of a domain of interest independent of specific applications in a general form and are used as semantically rich schemas for knowledge graphs. The W3C technology stack for ontologies consists of the Resource Description Framework (RDF)[14] and its Schema extension (RDFS)[15] that form the representational basis for the Web Ontology Language (OWL)[16], by which domain knowledge can be expressed in terms of logical statements that support automated reasoning for inferring implicit knowledge. Additional technologies are SPARQL[17] for querying and SHACL[18] for validating RDF-based data models.

In relation to the CSS Model proposed in this paper, Semantic Web Ontologies are a good choice of technologies for implementing the conceptual model specification from Chapter 3.3 in a computational form independent of specific programming languages and separated from the applications using it. By means of an RDF-based knowledge graph infrastructure, as e.g. provided by established triple stores, an OWL-based implementation of the CSS Model can be directly instantiated in applications in order to query, validate and reason over CSS-relevant application data. Applications that make use of skills, capabilities and services could harvest some immediate benefits such as RDF's unique global identification mechanism

---

[13] https://www.w3.org/
[14] https://www.w3.org/RDF/
[15] https://www.w3.org/TR/rdf-schema/
[16] https://www.w3.org/OWL/
[17] https://www.w3.org/TR/rdf-sparql-query/
[18] https://www.w3.org/TR/shacl/

through Internationalized Resource Identifiers (IRI), RDFS taxonomies for e.g. capability hierarchies and their recursive traversal when querying, plus OWL-based reasoning for the realization of matching techniques. Moreover, the rich tooling and community support for Semantic Web Technologies also enables immediate use.

There is already a plethora of literature in which especially the concepts of capability and skill have been taken up for realization with ontologies and Semantic Web standards, often with mixed or interchangeable wording for the two. An incomplete list is (Björkelund, Bruyninckx, Malec, Nilsson, & Nugues, 2012), (Järvenpää, Siltala, & Lanz, 2016), (Köcher, Hildebrandt, Vieira da Silva, & Fay, 2020), (Perzylo, et al., 2019), . (Weser, Bock, Schmitt, Perzylo, & Evers, 2020).

When applied in the context of digital twins and the related AAS, such an ontology-based implementation of the CSS Model can also serve as a semantic basis for a CSS-related AAS submodel template to which respective submodels can refer to by using semantic IDs.

5.3 Module Type Package

The *Module Type Package* (MTP) is a description of the interfaces and functions of modular units in process manufacturing modelled using AutomationML[19]. Executable functions are defined as services, which can be further subdivided into different procedures. Modules make their services available to a higher-level *Process Orchestration Layer* so that an aggregate process may be composed using services of multiple modules.

Standardization of the MTP has been carried out for several years by the organizations VDI/VDE, NAMUR and ZVEI in the VDI guideline series 2658[20] (Automation engineering of modular systems in the process industry: General concept and interfaces, 2019). While the CSS Model is mostly a conceptual model with a variety of individual proof-of concept implementations, the MTP standard series and the tools presented so far have already reached a higher level of maturity. The MTP can be seen as either an alternative approach to the CSS Model or as a specialized manifestation of the CSS Model, originating from process industries and targeting aspects typical for this type of production. Service interfaces and other aspects such as HMIs and alarms are defined and standardized very precisely.

Thanks to this standardization, MTP service interfaces can serve as skill interfaces according to the CSS Model. However, a semantically rich description of the capabilities and services

[19] https://www.automationml.org/

[20] https://www.vdi.de/richtlinien/details/vdivdenamur-2658-blatt-1-automatisierungstechnisches-engineering-modularer-anlagen-in-der-prozessindustrie-allgemeines-konzept-und-schnittstellen

– which is required for robust production planning methods or the shared production scenario – is missing in the MTP due to a lack of formal semantics in the MTPs description language.

There is current research work on an alignment of the MTP with discrete manufacturing in general and capabilities / skills in particular. The applicability of the MTP to manufacturing logistics is analyzed in (Blumenstein, Fay, Beers, Stutz, & Maurmaier, 2022). Furthermore, there are also initial works on a mapping between the MTP and a capability and skill model (Köcher, Beers, & Fay, A Mapping Approach to Convert MTPs into a Capability and Skill Ontology, 2022).

5.4 OPC Unified Architecture

OPC UA (OPC[21] Unified Architecture)[22] is a communication framework developed by the OPC Foundation with the aim of enabling manufacturer-independent data exchange. OPC UA has the potential to become a de facto Industrie 4.0 standard to connect the resources (production modules) in the field level. OPC UA is structured as a multi-layer communication framework. On the one hand, it supports communication via client-server as well as via publish-subscribe (PubSub) and supports various communication technologies. Aspects such as security and authentication are already taken into account in the framework. A big advantage of OPC UA is the possibility to describe information models. Standardized information models (Companion Specifications) are defined both by the OPC Foundation and by partners, e.g., in the VDMA (Verband Deutscher Maschinen- und Anlagenbau e.V.) in cross-company working groups.

Over the course of the last years, implementation possibilities of skills have been investigated within the scope of various research projects (Hammerstingl, Reinhart, & Zimmermann, 2016), (Dorofeev & Zoitl, 2018), (Zimmermann, et al., 2019), (Volkmann, 2021). While the actual implementation of the skill still has to be realized in a resource-specific way by proprietary control code, the communication standard OPC UA has proven to be a promising approach for the implementation of the skill interface. The description of the skill with all its properties and parameters can be done directly within an information model in the OPC UA server. OPC UA also offers various interaction mechanisms to not only monitor skills, but also to enable controlling access to them. The possibility of using OPC UA for offering a skill interface is discussed in (VDMA, 2022).

[21] Today the acronym OPC stands for Open Platform Communications. Source: What is OPC? - OPC Foundation accessed: 26.07.2022

[22] https://opcfoundation.org/about/opc-technologies/opc-ua/

5.5 Automation Markup Language

AutomationML is an object description language including an XML-based file format for modeling extensive engineering information, it is used for data exchange between different engineering tools. AutomationML is specialized in object models, geometry and behavior description, supports libraries, versioning and was developed for the iterative exchange of engineering data.

Capabilities, skills, and services of production systems are modelled in AutomationML in relation to the product-process-resource context which is included in the standard (Schleipen & Drath, 2009), (Pfrommer, Schleipen, & Beyerer, 2013). This serves as basis for use cases driven by engineering such as Plug and Produce. Specific Role Classes which describe semantics within AutomationML may be created to indicate that an element within the model is a skill, a capability, or a service. These Role Classes may be enriched with specific attributes and constraints. The Role Classes may be collected in a common Role Class Library, Attributes may be collected in a common Attribute Type Library. Currently, there are no standardized libraries for the CSS Model, but project-specific realizations of this concept.

5.6 Service-Oriented Architecture

A service-oriented architecture (SOA) is an architectural paradigm that encourages using multiple services to structure software functionality which may be distributed and maintained by different owners (OASIS, 2006). Services are typically considered to be self-contained functions that may be composed of other services and logically represent a recurring activity with a clearly defined input and outcome so that consumers of the service may interact with it in the sense of a “black box” - i.e. without knowing a service’s internal details (The Open Group, 2009).

What should be noted is that the understanding of the term “service” as used in information technology is significantly different from the understanding expressed in this document. While services in IT are encapsulated functionalities and may thus be compared to skills, services in the context of the CSS model act as containers bundling capabilities with commercial aspects in order to be offered and requested on a marketplace (see Section 3.3.4).

In fact, transferring the SOA service concept from information technology to automation was one of the earliest approaches to obtaining encapsulated functions with clearly defined interfaces in automation (Jammes & Smit, 2005). Thus, services can be seen as an early precursor of skills according to the CSS Model.

5.7 Related Standards

Next to the previously listed approaches and technologies that primarily originate from the area of Information Systems and Computer Science, there is a plethora of standards applied in various industries, where recognized standardization organizations have established consensus on guidelines across consortia of industrial partners. Many of these established standards can play an important role when concretizing the application of capabilities, skills and services in an industrial domain.

Specific standards that may be considered in relation to the capability part of the CSS Model are DIN 8580 (DIN Deutsches Institut für Normung e. V., 2020), VDI 2860 (VDI Verein Deutscher Ingenieure e.V., 1990) and TGL 25000 (Verfahrenstechnik Grundoperationen, TGL 25000, 1974). These standards define terminology for different domains, which can be helpful to classify the terms of capabilities. DIN 8580 defines a taxonomy of manufacturing methods, which is helpful for the description and classification of capabilities. The guideline VDI 2860 provides a classification, delimitation and definition in the solution of handling tasks as well as in its sub-functions. The TGL 25000 describes basic process engineering operations for the production of formless materials, which are gases, liquids, pastes, powders, granulates and similar materials. Basic operations of process engineering are target-oriented actions which change the qualitative or quantitative composition, the degree of distribution or the energy content by means of physical processes in the material to be treated.

The IEC Common Data Dictionary (CDD[23]) serves as a common repository of concepts for all industrial/technical domains (electrotechnical and non-electrotechnical; e.g. industry, building, energy, healthcare, …) based on the methodology and the information model of IEC 61360 series. The textual definitions in the CSS Model have already been aligned to CDD terminology in large parts, and CDD can also serve the further concretization of extended terminology in model specializations for specific target industries.

EClass[24] is a global reference data standard for the classification and unambiguous description of products and services. It can specifically serve the purpose to provide supplementary vocabulary for classes (types) and their properties to extend the product dimension of the CSS Model.

[23] IEC 61360-4 - IEC/SC 3D - Common Data Dictionary (CDD - V2.0015.0002)

[24] EClass https://eclass.eu/

6 Outlook

This document introduces an information model for capabilities, skills and services in order to improve the common understanding of these elements and their relation in the area of new production concepts as well as to provide a basis for upcoming standardization activities.

Furthermore, application scenarios and challenges which can be addressed with capabilities, skills and services are presented. Generic use cases and two application examples describe the context of application of the CSS model in more detail.

We see these use cases and examples as a starting point of activities where the information model can serve as base for specializations in specific application areas or industries and give impetus for development and research activities. For that purpose, the document gives a basic overview of possible implementation technologies (e.g. AAS, OPC UA information models, Semantic Web Technologies, BaSys Control Components), which today are already used for realization of capability and skill-based systems. Having a common basis will help to focus development efforts and improve the interoperability of future solutions and products.

As standardization will be a crucial aspect for some applications, particularly in shared production scenarios, and a common approach for modeling is a prerequisite for successful development of standards, we hope that the CSS model can deliver a significant contribution. One example for a corresponding activity is the alignment with the ongoing work within the new IDTA Capability submodel working group.

OPC UA is an important technology in the manufacturing an automation domain. Therefore, the VDMA has launched a conceptual paper how to execute skills using OPC UA interface implementations. This conceptual paper is compatible with the CSS model which is introduced in this white paper [VDMA (2022)].

We want to encourage readers of this document to give feedback and use the CSS Model for further developments to leverage the full potential of the underlying concepts and technologies within future industrial production systems.

7 Literature

[VDI/VDE/NAMUR 2658-1]. (kein Datum).

Automation engineering of modular systems in the process industry: General concept and interfaces. (10 2019). *VDI/VDE/NAMUR 2658-1*.

Björkelund, A., Bruyninckx, H., Malec, J., Nilsson, K., & Nugues, P. (2012). Knowledge for Intelligent Industrial Robots. In: George Konidaris. *AAAI Technical Report SS-12-02, Designing Intelligent Robots: Reintegrating AI*, S. SS-12-02.

Blumenstein, M., Fay, A., Beers, L., Stutz, A., & Maurmaier, M. (J 2022). Vergleichende Untersuchung der Orchestrierung modularer Anlagen in der Prozessindustrie, der Fertigungsindustrie und der produktionsnahen Logistik. *Automation 2022: 23. Leitkongress der Mess- und Automatisierungstechnik: automation creates sustainability*.

Borgo, S., Terkaj, W., & Sanfilippo, E. (2021). Capabilities, Capacities, and Functionalities of Resources in Industrial Engineering. *CEUR Workshop Proceedings of the 11th International Workshop on Formal Ontologies meet Industry*.

DIN Deutsches Institut für Normung e. V. (2020). Fertigungsverfahren - Begriffe, Einteilung, DIN 8580.

Dorofeev, K., & Zoitl, A. (2018). Skill-based Engineering Approach using OPC UA Programs. *IEEE 16th International Conference on Industrial Informatics (INDIN)*, S. 1098-1103.

Hammerstingl, V., Reinhart, G., & Zimmermann, P. (2016). Automatisierte Konfiguration und Selbstauskunft von Industrierobotern. *Industrie 4.0 Managment*.

ISO. (2004). ISo15531-31: Industrial automation systems and integration - Industrial manufacturing management data - Part 31: Resource information model.

Jammes, F., & Smit, H. (2005). Service-Oriented Paradigms in Industrial Automation. *IEEE Transactions on Industrial Informatics, Vol. 1 (1)*, S. 62–70.

Järvenpää, E., Siltala, N., & Lanz, M. (2016). Formal Resource and Capability Descriptions Supporting Rapid Reconfiguration of Assembly Systems. *IEEE International Symposium on Assembly and Manufacturing (ISAM)*, S. 120–125.

Köcher, A., Beers, L., & Fay, A. (2022). A Mapping Approach to Convert MTPs into a Capability and Skill Ontology. *accepted for publication at ETFA 2022*.

Köcher, A., Hildebrandt, C., Vieira da Silva, L., & Fay, A. (2020). A Formal Capability and Skill Model for Use in Plug and Produce Scenarios. *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, S. 1663–1670.

OASIS. (2006). Reference Model for Service Oriented Architecture. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.

Perzylo, A., Grothoff, J., Lucio, L., Weser, M., Malakuti, S., Venet, P., Deppe, T. (2019). Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective. *IFAC-PapersOnLine, Vol. 52 (13)*, S. 1590–1596.

Pfrommer, J., Schleipen, M., & Beyerer, J. (2013). PPRS: Production skills and their relation to product, process, and resource. *Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*.

Plattform, A. (2016). *Fortschreibung der Anwendungsszenarien der Plattform Industrie 4.0.* Berlin: Bundesministerium für Wirtschaft und Energie (BMWi), Öffentlichkeitsarbeit.

Roman Froschauer, A. K. (2022). Capabilities and Skills in Manufacturing: A Survey Over the Last Decade of ETFA. *IEEE Conference on Emerging Technologies & Factory Automation (ETFA)* .

Sarkar, A., & Sormaz, D. (2019). Ontology model for process level capabilities of manufacturing resources. *Procedia Manufacturing 39*, S. 1889-1898.

Schleipen, M., & Drath, R. (2009). Three-View-Concept for modeling process or manufacturing plants with AutomationML. *IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, S. 1-4.

Solano, L., Romero, F., & Rosado, P. (2016). An ontology for integrated machining and inspection process planning focusing on resource capabilities. *International Journal of Computer Integrated Manufacturing 29*, S. 1-15.

The Open Group. (2009). *The SOA Source Book.* Available at: http://www.opengroup.org/soa/source-book/intro/index.htm.

VDI Verein Deutscher Ingenieure e.V. (1990). Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole, VDI 2860.

VDMA. (2022). VDMA Leitfaden. *„Fähigkeiten in der Produktionsautomatisierung - Konsolidierung des Konzepts aus Sicht des Maschinen- und Anlagenbaus mit dem Schwerpunkt OPC UA. VDMA 2022.*

Verfahrenstechnik Grundoperationen, TGL 25000. (1974). *VVB Chemieanlage*.

Volkmann, M. (2021). Integration of a feasibility and context check into an OPC UA skill. *IFAC-PapersOnLine 54.1*, S. 276-281.

Weser, M., Bock, J., Schmitt, A., Perzylo, A., & Evers, K. (2020). An Ontology-based Metamodel for Capability Descriptions. *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, S. 1679–1686.

Zimmermann, P., Axmann, E., Brandenbourger, B., Dorofeev, K., Mankowski, A., & Zanini, P. (2019). Skill-based Engineering and Control on Field-Device-Level with OPC UA. *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, S. 1101-1108.

8 Annex A – Terms

8.1 Capability

implementation-independent specification of a function in industrial production to achieve an effect in the physical or virtual world.

Notes:

- *A capability can be restricted by constraints.*
- *A capability can be specified by capability properties.*
- *A capability can be realized by skills.*

8.2 Property

quality or characteristic inherent in or ascribed to any CSS model element

Notes:

- *Properties may be used to describe and differentiate all kinds of PPR entities (i.e., products, process steps, resources)*
- *Capabilities, services and offers are specified by properties in order to detail their description with regard to certain entities (e.g. products, process steps, resources)*

Sources:

- *Adopted from IEC 61360-1:2017 (Properties)*

8.3 CapabilityConstraint

condition imposed on a capability that further details its applicability.

Notes:

- *A capability constraint can be formulated as one of the following three constraint types:*
  - *A precondition, i.e., a condition that must hold before a function can be executed.*
  - *A postcondition, i.e., a condition that must hold after a function has been executed.*
  - *An invariant, i.e., a condition that must hold during the execution of a function.*

- *A capability constraint restricts the values of the properties associated with the respective capability.*
- *A capability constraint can involve one or more properties.*

8.4 Service

description of the commercial aspects and means of provision of offered capabilities.

Notes:

- *The term "service" should be understood in the sense of economics and shall not be confused with e.g., web services.*
- *The capabilities and means of provisions are specified by properties*
- *A service is demanded by service requesters and provided by a service provider*
- *A service is an input for an offer proposed by a service provider, which can be received and accepted by a service requester*

8.5 ServiceRequester

demands services under particular commercial aspects by providing either a specification of services or a specification of product requirements.

8.6 ServiceProvider

provides services and can propose offers to ServiceRequesters.

8.7 ServiceOffer

proposal for a binding contract from the Service Provider to execute one or more particular services that a ServiceRequester can receive and accept.

Notes:

- A ServiceOffer should determine the commercial aspects of the service provision and may remain valid for a certain period of time.
- An Offer may consist of partial offers proposed by different service providers.

8.8 Skill

executable implementation of an encapsulated (automation) function specified by a capability.

Notes:

- A skill must have a skill interface
- One capability can be realized by more than one skill.
- A skill may have any number of SkillParameters.

- A skill's behavior conforms to a state machine.
- A skill controls a process step

8.9 SkillInterface

access point to configure, control and monitor a skill.

Notes:

- A skill interface exposes interaction points to be used by other external systems (e.g. MES, other skills).
- A skill interface exposes the state machine of a skill so that skill states can be monitored and transitions triggered.
- A skill interface exposes the parameters of a skill so that they can be written and read.

8.10 SkillParameter

data unit to configure, control and monitor the execution of a skill.

Notes:

- Skill parameters might be used as in- /output parameters
- Skill parameters might be used as results
- Skill parameters might have a relation or be equivalent to capability properties

Source:

IEC 171-05-41

Examples:

- Skill parameter that is equivalent to a capability property: color of product
  - Property value is required by service requester
  - Property specifies a provided capability and may be used in capability constraints
  - Property value is passed to a skill parameter
- Skill parameter that is indirectly determined from a capability property:
  - Property: product material
  - Skill parameter: feed rate
  - Property value is required by service requester
  - Property specifies a provided capability and may be used in capability constraints
  - Skill parameter value needs to be calculated from property value before passing it to a skill

8.11 Process

production-relevant activity at any level of granularity that might affect materials and is performed by resources

Notes:

- In general, a process can be decomposed into sub-processes or single activities.
- A *process* can require *capabilities* to express that any suitable *resource* used for performing this *process* needs to provide compatible *capabilities*.
- A process step relates to materials that constitute either input or output for the processing in this step.

Source:

Specialized from Process according to IEV 351-42-33 (https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-42-33) so that a single activity may also be considered as a process.

Examples:

Process step range from a subprocess for e.g. putting a complex assembly together, down to a single activity, such as fixating a screw. Moreover, also logistics-related actions are examples of process steps, including the feeding of materials into a working unit, the storage of intermediate products in a buffer, or the transportation of materials in between different factory facilities. Furthermore, also setup actions are considered process steps, such as the preparation of a machining station with the right tool, the mounting of the right gripper to a robot, or the cleaning of the working area in a station.

8.12 Resource

entity capable of performing functions specified as capabilities and potentially implemented as skills. [compare definition of "Functional Unit"]

Notes

- A production resource may consist of hardware, software or both
- A production resource may only provide capabilities (i.e. when engineering a resource, for planning purposes) and may additionally provide skills for automatic execution of the specified function.
- A human becomes a Production Resource, if that person is able to perform a function specified as a capability.

Sources:
- IEV 171-01-22

8.13 Product

physical object being used as an input or created as an output of a production process.

Notes:

- The term *Product* may be used for objects in various states of manufacturing and may be seen as a generic term for raw materials, work in process and finished goods. Consumable supplies such as fuel, lubricants or cleaning agents may also be regarded as products. Furthermore, both purchased parts as well as parts manufactured in-house may be regarded as products.
- Besides the actual product, there can be additional artifacts related to that product that are created and used in different life cycle phases to specify the product.
  - 3D/CAD models
  - specifications
  - BOM
- A Service Requester may use these additional artifacts to formulate requirements against products.

Source:

This notion of a product is derived from IEV 902-02-03 and specialized to capture physical objects only.

9 Annex B – Activities

Note: Activities are presented in the same order in which they are used in the use-case.

9.1 ServiceSequencing

activity of eliciting the required services and deriving the sequence of services needed to achieve a desired effect required by one or more products in a particular state to transition to one or more products in a subsequent state.

9.2 ServiceMatching

activity of comparing and assigning requested services and tender criteria to assured services and creating a binding offer to a service requester.

Notes:

- That includes non-process related commercial properties (e.g. supply chain Law)
- ServiceMatching uses the activity of CapabilitySequencing. While the CapabilitySequencing can be performed without the ServiceMatching, the ServiceMatching depends on the CapabilitySequencing.
- A prerequisite is agreed semantics for the definition of the services and properties.

9.3 ComplianceCheck

activity of verifying adherence of the ServiceProvider to a set of rules or laws, such as relating to policies or product and business standards and codes of conduct.

Notes:

- ComplianceCheck may be carried out by a third party.

Examples: Compliance with the Supply Chain Law, country embargos, minimum wage, environmental safety, CE mark, non-use of child labor, fair trade, non-use of certain substances such as toxins and carcinogens.

9.4 CapabilitySequencing

activity that may be used to find a sequence of capabilities which lead from a given initial state to a target state of a product. *CapabilitySequencing* may contain *RequiredCapabilityDerivation, CapabilityMatching* and *CapabilityDecomposition*.

Notes:

- CapabilitySequencing may include optional optimization steps to achieve a specific goal (e.g. minimum cost or maximum throughput)
- CapabilitySequencing may result in an incomplete fulfillment of the product requirements so that product requirements must be altered or additional service providers must be found

9.5 RequiredCapabilityDerivation

activity of eliciting the necessary capabilities to achieve a desired effect required by one or more products in a particular state to transition to one or more products in a subsequent state.

Notes:

- If specific capabilities are requested instead of requested product properties, this activity may be omitted.

9.6 CapabilityMatching

activity of identifying the resources and their provided capabilities that fulfill required capabilities. This involves ensuring that a given set of CapabilityConstraints is fulfilled.

Notes:

- A prerequisite is agreed semantics for the definition of the Capabilities, Properties and Constraints. Unknown CapabilityConstraints or Properties should lead to a warning about the result.
- CapabilityMatching may be done iteratively with adjustment of the required CapabilityConstraints depending on the result of the Capability Matching. Moreover, adjustment of the CapabilityConstraints could be part of a negotiation activity at service level.
- The resources to be identified may be individual assets such as factories or machines or groups of such assets, forming for example a supply chain.

9.7 CapabilityDecomposition

activity of breaking down (decomposing) a capability into subordinate or part capabilities that together fulfill the function of the original capability.

Notes:

- CapabilityDecomposition may be applied hierarchically over multiple levels with increasing refinement, e.g. corresponding to an hierarchy of manufacturing processes and subprocesses.
- CapabilityDecomposition defines the order of the subordinate or part capabilities either explicitly or implicitly using the Capability Constraints (Pre- and Postconditions)
- CapabilityDecomposition can be used to allow the matching of provided and requested capabilities if both are not specified at the same level of granularity
- CapabilityDecomposition implies that instead of breaking down a capability, a set of subordinate or part capabilities may also be composed into a higher level capability.

9.8 FeasibilityCheck

activity to assess the possibility to achieve the desired effect of a skill execution in a concrete production context.

Notes:

- The FeasibilityCheck result can be used as assurance or validation to execute a process
- The FeasibilityCheck may use skill specific information such as StateMachine, SkillParameter, SkillInterface
- In addition to skill information the following information of a production context may be used:
    - Topology
    - Environmental (contextual) model of the factory plant

  - Simulation model
  - Mathematical-physical formulas etc.
- There can be multiple options to realize FeasibilityChecks, e.g.:
  - Parameter matching at the product resource level.
  - Mathematical-physical calculations
  - Simulation
  - Domain specific (e.g. welding)
- The Feasibility Check typically receives the following as inputs:
  - Process flow with one or more production resources and their required capabilities.
  - Feasibility model of the product resource(s)
- The result can be a specific configured production resource and/or a process with parameterized skill as well as costs and other process-related commercial information
- The FeasibilityCheck might include activities to assure the required resource capacity and generate an offer

Authors

Christian Diedrich, ifak e.V. Magdeburg

Alexander Belyaev, Otto von Guericke University

Rafi Blumenfeld, Siemens AG

Jürgen Bock, Technische Hochschule Ingolstadt

Stephan Grimm, Siemens AG

Jesko Hermann, Technologie-Initiative SmartFactory KL e.V.

Tobias Klausmann, Lenze SE

Aljosha Köcher, Helmut Schmidt University

Matthias Maurmaier, Siemens AG

Kristof Meixner, TU Wien

Jörn Peschke, Siemens AG

Miriam Schleipen, EKS InTec GmbH

Siwara Schmitt, Fraunhofer IESE

Boris Schnebel, Fraunhofer IOSB

Guido Stephan, Siemens AG

Magnus Volkmann, TU Kaiserslautern

Andreas Wannagat, Siemens AG

Kym Watson, Fraunhofer IOSB

Michael Winter, RWTH Aachen University

Patrick Zimmermann, Fraunhofer IGCV

This publication was created by the working group "Semantic and Interaction of I4.0 Components" of Plattform Industrie 4.0.