at – Automatisierungstechnik 2023; 71(2): 94–104 DE GRUYTER OLDENBOURG

Methods

Aljosha Köcher*, Alexander Belyaev, Jesko Hermann, Jürgen Bock, Kristof Meixner, Magnus Volkmann, Michael Winter, Patrick Zimmermann, Stephan Grimm and Christian Diedrich

A reference model for common understanding of capabilities and skills in manufacturing

Ein Referenzmodell für ein gemeinsames Verständnis von Capabilities und Skills von Anlagen

https://doi.org/10.1515/auto-2022-0117
Received September 15, 2022; accepted November 21, 2022

Abstract: In manufacturing, many use cases of Industrie 4.0 require vendor-neutral and machine-interpretable information models to describe, implement and execute resource functions. Such models have been researched under the terms *capabilities* and *skills*. Standardization of such models is required, but currently not available. This paper presents a reference model developed jointly by members of various organizations in a working group of the *Plattform Industrie 4.0*. This model covers definitions of most important aspects of capabilities and skills. It can be seen as a basis for further standardization efforts.

Keywords: capabilities; resource functions; services; skill; standardization.

Zusammenfassung: Für viele Anwendungsfälle von Industrie 4.0 werden herstellerneutrale und maschinenlesbare Informationsmodelle von *Capabilities* und *Skills* benötigt. In der Praxis sind standardisierte Modelle erforderlich, welche zur Zeit jedoch nicht existieren. In diesem Beitrag wird ein Referenzmodell vorgestellt, das von Mitgliedern verschiedener Organisationen in einer Arbeitsgruppe der *Plattform Industrie 4.0* erarbeitet wurde. Dieses Modell umfasst Definitionen der wichtigsten Aspekte von Capabilities und Skills und kann somit als wichtige Grundlage für weitere Standardisierungsarbeiten angesehen werden.

Schlagwörter: Capabilities; Maschinenfunktionen; Services; Skills; Standardisierung.

1 Introduction

A substantial trend in future manufacturing is the requirement for a faster reaction to market uncertainties demanding more flexibility in industrial production [1]. This flexibility concerns many different aspects, e.g., the ability to quickly introduce new products or product variants. Therefore, the potential to efficiently produce high-mix scenarios under low volume or even *lot size one* is necessary. These challenges require new concepts for production control and the ability to react to problems and disturbances within production and supply chains. One possible approach to tackle this challenge is modularizing production resources and requirements and abstracting them to the functions requested or provided. However, the information on available and provided functions must be communicated to the interaction partners so that all interaction partners maintain a common understanding.

In recent years, many research activities have taken place to develop the needed concepts in detail and elaborate their application in the industrial domain.[1] Accordingly, a wide variety of terms and concepts have emerged that are relevant for describing functionalities of assets [2].

1 For instance, the Conference on Emerging Technologies and Factory Automation (ETFA) hosts a special session on this topic.

***Corresponding author: Aljosha Köcher**, Helmut-Schmidt-Universität, Hamburg, Germany, E-mail: aljosha.koecher@hsu-hh.de
Alexander Belyaev and Christian Diedrich, Otto von Guericke Universität, Magdeburg, Germany,
E-mail: alexander.belyaev@ovgu.de (A. Belyaev), christian.diedrich@ovgu.de (C. Diedrich)
Jesko Hermann, Technologie-Initiative SmartFactory KL e.V., Kaiserslautern, Germany, E-mail: jesko.hermann@smartfactory.de
Jürgen Bock, Technische Hochschule Ingolstadt, Ingolstadt, Germany, E-mail: juergen.bock@thi.de
Kristof Meixner, Christian-Doppler-Laboratory-SQI, TU Wien, Vienna, Austria, E-mail: kristof.meixner@tuwien.ac.at
Magnus Volkmann, TU Kaiserslautern, WSKL, Kaiserslautern, Germany, E-mail: magnus.volkmann@mv.uni-kl.de
Michael Winter, RWTH Aachen University, Aachen, Germany, E-mail: m.winter@plt.rwth-aachen.de
Patrick Zimmermann, Fraunhofer-Institut für Gießerei-, Composite- und Verarbeitungstechnik IGCV, Augsburg, Germany, E-mail: patrick.zimmermann@igcv.fraunhofer.de
Stephan Grimm, Siemens AG, Munich, Germany, E-mail: stephan.grimm@siemens.com

In today's discussion in the Industrie 4.0 community, various similar terms, e.g., "capability", "skill" or "task", are used to describe a function of an asset. However, to the best of our knowledge, a clean differentiation of these terms does not exist. Often, different authors use various terms to describe similar concepts. Vice versa, the same term is assigned to contrary concepts. Furthermore, functions need to be communicated on different levels of abstraction for use cases ranging from automated order handling in supply chains to manufacturing execution.

All these issues complicate the comparison of existing approaches and lead to confusion. Accordingly, the emerging solutions cannot be interoperable per se. This work aims to consolidate the different company- and institution-specific terms by answering the following research questions.

- What is the essential set of concepts from flexible production that must be covered to communicate available and required functions?
- What does a model look like that relates these concepts?
- What are potential technical implementations of this model?

As a result, we present the Capability-Skill-Service (CSS) model, which was jointly developed in a working group of *Plattform Industrie 4.0* and thus represents a consensus between different companies and research institutes. The CSS model can be seen as the first reference model of capabilities, skills, and services, defining and categorizing them for a clear distinction and specifying their relationships with each other. Furthermore, an overview of technology mappings that are currently being researched for each aspect of the model is presented. Lastly, this contribution compares the CSS model with existing, similar approaches.

The remainder of the paper is structured as follows: Chapter 2 gives an overview of related activities and research work. Chapter 3 introduces the CSS reference model with extended definitions and explanations of the concepts *capability*, *skill*, and *service* before Chapter 4 presents potential technology-specific implementations. Chapter 5 summarizes approaches that pursue similar goals, i.e., the encapsulation of automation functions and description of their interfaces. Finally, Chapter 6 recaps and critically evaluates the CSS reference model and provides an outlook of possible future research activities.

2 Related work

In production systems engineering, organizations work in multidisciplinary engineering environments, where stakeholders maintain different views on the manufactured products and the production system. The Product-Process-Resource (PPR) concept, described in [3], represents the three major aspects of production systems engineering. *Products* represent input and output products, *processes* describe production processes that transform input into output products, and *resources* describe production resources that execute the processes. The Formalized Process Descriptions (FPD), defined in VDI 3682 [4], provides a visual and formal model to describe these aspects.

Pfrommer et al. [1] introduced *skills* as additional element to the PPR concept. The authors defined skills as vendor-independent representations of production process functionality required by a product and provided by a resource. These characteristics enable the abstraction between production processes and resources.

A more thorough terminological discussion of functionality and function related to capabilities (and even capacities) can be found in [5]. The authors of [5] also include further references on formal terminological clarification of *function* in the ontology engineering community.

Earlier publications often use the concepts of capabilities and skills interchangeably [2]. However, there is a more apparent distinction between capabilities and skills in recent literature. While capabilities are often defined as an abstract description of a function provided by a machine, skills are typically seen as executable implementations of these functions that might be used to execute a process on a machine. Even though this distinction is slowly emerging, there is no holistic, integrated model of PPR on one side and capabilities and skills on the other side so far [2].

The *Plattform Industrie 4.0* describes the individual generation of processes based on product descriptions and defined capabilities. capabilities are defined as a vendor-neutral description of functions, while a skill is an implementation of a resource to realize a function [6]. Possible technologies, e.g., to model capabilities in ontologies or different realizations of skills, are discussed [6]. Another level of abstraction comes to this discussion around capabilities and skills when talking about supply chains spanning across company borders. For such a network with service providers to share production resources, [7] defines *cloud manufacturing* as "a model for enabling aggregation of distributed manufacturing resources […] to a shared pool of configurable manufacturing services that can be rapidly provisioned and released with minimal management effort or service operator and provider interaction".

While standards around capabilities, skills and services do currently not exist, a selection of standards is regularly incorporated into models. In addition to VDI 3682 [4], most often, standards for the definition of process types such as DIN 8580 (manufacturing processes) or VDI 2860 (handling) are used (e.g., [8, 9]). Furthermore, the state machine defined in *PackML*/ISA 88 is often used to model the behavior and interactions of skills (e.g., [8, 10]).

3 Capability-Skill-Service model

As an attempt to consolidate the terminology around capabilities, skills and services for manufacturing, the conceptual model presented in this section extends the well-established PPR paradigm [3] by the additional notion of a *manufacturing function*. The PPR paradigm focuses on a *process* as the main element modeled for production. *Products* are the outputs of a production process. Similarly, raw materials or semi-finished parts are also considered *products*, which are the inputs to processes. A *resource* represents a machine or plant that provides the functionality to execute processes. Additionally, PPR elements may have *properties* that can characterize their instances according to [11]. Furthermore, each element has a reflexive relation that allows the elements to be self-composed, e.g., a process may be composed of sub-processes or a product of product parts. The PPR concepts with their properties are illustrated in Figure 1 on the left-hand side. The relation between the PPR concepts and the properties is not shown explicitly to prevent cluttering the figure. In the initial PPR approach [3] and the VDI 3682 [4], processes and resources are bound by a direct *usage* relation, making the model relatively rigid.

However, as manufacturing systems engineering requires information modeling at different levels of system functionality, it is not sufficient to simply add *function*[2] as an element to PPR. Instead, the notion of function needs to be reflected at all relevant levels of the functional hierarchy defined in [12]: An abstract description of a factory's functionality is needed on *levels* 3 and 4 (manufacturing operations management and business planning) for production planning purposes. The invocation of functions is the responsibility of *levels* 2 (supervisory control) and results in control operations on *level* 1. Externally offering functionalities to form shared production networks concern *levels* 4 and higher levels. Thus, the otherwise rigid relation between processes and resources needs to be decoupled by a separate description of required and provided functionality. To this end, we introduce three model elements which extend PPR, all representing aspects of function for different uses. These are the *capability* as an abstract description of a function, the *skill* as an invocable function, and the *service* as a function offering to external partners on the level of dynamically integrating and connecting processes.

After a short discussion of requirements and an overview of the model, the three concepts *capability*, *skill*, and *service* are presented in separate subsections.

3.1 Requirements and model overview

An overview of requirements from various publications was condensed by [2]. The most relevant requirements for this work are discussed here, while more specific requirements, e.g., regarding solution technologies, are out of the scope of this work. Models of capabilities, skills, and services need to foster more efficient approaches to production planning and production system reconfiguration. This need can be considered a paramount requirement from which all others can be derived. Both *matchability* and *executability* are often mentioned as a requirement highlighting the need for a description of functions on different levels [2]. While matchability may best be achieved with formal models, executability necessitates bindings to implementation technologies. Thus, a clear distinction between these concepts is needed. Skills must have a *communication interface*, and individual skill *states* need to be expressed [2].

Figure 1 represents a Unified Modeling Language (UML) class diagram to illustrate the developed CSS reference model considering the previously discussed requirements. The model distinguishes the three areas capability, skill, and service. In each of the areas, the main concepts and their relations to other model elements are defined. In addition, the main concepts of each aspect are related to PPR concepts.

3.2 Services

One of the promising and future-oriented developments of the manufacturing industry is the upcoming transformation of industrial production into shared production. According to this scenario, production sites will form cross-company networks. In such networks, *service providers* can offer their manufacturing capabilities and integrate the capabilities of external partners into their own production processes based on specific orders. Such scenarios imply automated order processing in supply chains spanning

2 We understand a *function* in a general sense, as, e.g., discussed in the ontology engineering literature [5]—the function of, e.g., a hammer is to drive in nails—rather than a mathematical or computer science sense.

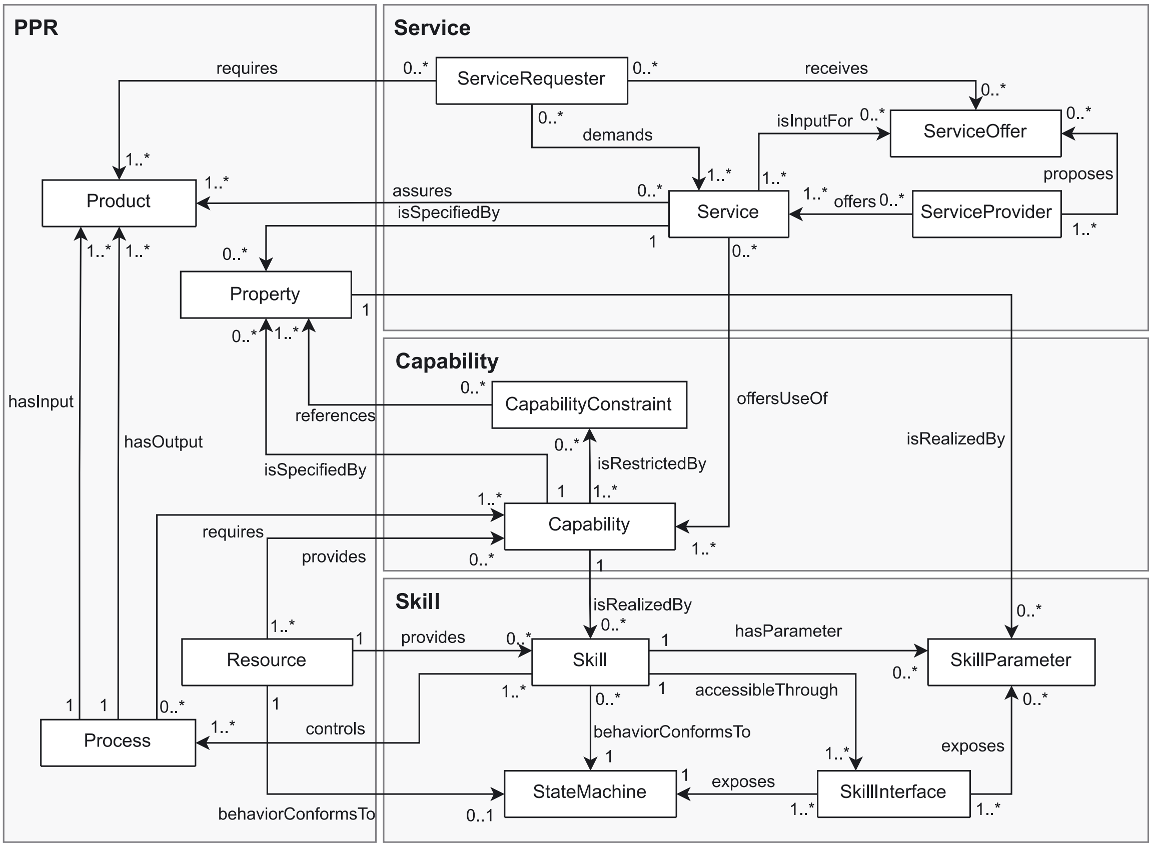

Figure 1: The reference model of capabilities, skills, and services with alignment to the PPR approach represented as a UML class diagram.

various companies. The challenge in realizing such shared production networks is interoperability. In this context, important parameters beyond a pure technical description of provided capabilities need to be considered for order processing [11, 13]. These are, for example, information on economic criteria such as delivery dates, cost, and agreements regarding documentation or maintenance, certification, and rating. For modeling such issues and distinguishing them from the concept of capability, this work introduces the term *service* in an economic sense representing a set of capabilities supplemented by an organizational and economic description. It should be noted that the service concept presented here is different from the same term used in information technology (see Section 5.1 for a distinction).

In the context of the CSS model, a *service requester*, which provides a specification of a requested service with its properties, demands suitable services. service providers can provide services suitable to fulfill a demanded service. They then can propose a service offer as the basis of a binding contract to execute one or more services. The service requester can accept the proposal under the proposed conditions in a specified time period. If service requesters are searching for services through a marketplace, multiple service offers may be created by different service providers. These service offers could be mutually exclusive or could also be combined to fulfill a requested service.

3.3 Capabilities

We define a *capability* as "an implementation-independent specification of a function in industrial production to achieve an effect in the physical or virtual world". Thus, a capability specifies a function in a production process. Usually, capabilities specify production functions that have an effect in the physical world. Nevertheless, software functions that only apply to the virtual world may also be modeled as a capability. capabilities that specify a production function should refer to terms of an actual manufacturing method, such as "drilling", along with properties and *capability constraints* to precisely describe their application. An example of a capability could read "drilling a hole with a particular depth and a diameter into certain types of material".

capabilities are either provided by production resources that claim the ability to apply the expressed function or are required by a process as part of a product's functional requirements. Required and provided capabilities typically differ, e.g., because provided capabilities are described in more detail to enable reuse for a range of operating conditions or even different processes. A matching between required and provided capabilities is thus necessary to find candidates for a suitable sequence of production steps for given requirements. This matching 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 the CSS model, capabilities are related to the concepts *skill* and *service*. The implementation of capabilities is possible through skills, which contain details at the level of implementation and invocation of automation functions. In a broader supply chain network outside a company-internal production setup, capabilities are offered through services.

3.4 Skills

We define a *skill* as "an executable implementation of an encapsulated (automation) function specified by a capability". A skill is provided by a resource in the production environment and enables the realization of a capability. Every capability may reference multiple skills in a production environment that act as implementations for this capability. skills must have a *skill Interface* allowing external systems, e.g., a Manufacturing Execution System (MES), to interact with the provided function. Every skills behavior needs to follow a harmonized *state machine* describing possible states and transitions. The state machine needs to be exposed via the skill interface so that the current state can be monitored and transitions can be triggered. The execution of one or more skills allows to control production steps. skills can have input or output *skill parameters* enabling the execution or monitoring of a skill. These skill parameters must also be exposed by the skill interface in order to set or get parameter values. Specifying input parameters makes it possible to execute defined production steps with an individual configuration. Every skill parameter references a property. On the one hand, this property may be a capability property, i.e., a property that is also used in a capability constraint. On the other hand, this property may also be a process or resource property which may only be relevant for execution but not for planning on capability level.

The distinction between capabilities and skills decouples the description of a function from its implementation and enables developers to freely select a technology and programming language to implement skills. In addition, multiple skill interfaces may be provided for one skill. This further decouples the skill implementation from its users that consume a skill only through its interface without being bound to the actual implementation. Integrators may select a skill interface matching their technology stack. With a suitable software architecture, skill interfaces can be installed or configured at the time of integration [14]. Admittedly, this requires more flexible and modular control approaches than the ones defined in IEC 61131.

4 Implementation of model elements

There is currently no standardization regarding the implementation of the CSS reference model, and existing approaches typically favor different technologies for different model aspects. This section presents implementation examples that are currently being researched for the three model aspects capabilities, skills, and services.

4.1 Modeling capabilities using ontologies

Semantic Web technologies provide mechanisms for knowledge representation in information systems. They are based on a stack of downward-compatible languages for information models and knowledge representation standardized by the W3C. Ontologies constitute reusable information models that capture the knowledge of a domain in a general form, independent of specific applications and are used as semantically rich schemas for knowledge graphs. The W3C technology stack for ontologies consists of the Resource Description Framework (RDF)[3] and its schema extension (RDFS)[4] that form the representational basis for the Web Ontology Language (OWL).[5] OWL allows to express domain knowledge in terms of logical statements that support automated reasoning for infering implicit knowledge. Additional powerful technologies such as SPARQL[6] and SHACL[7] may be used to query for and validate the information in RDF-based data models.

Semantic Web technologies provide an ideal candidate solution for the semantically rich representation of capabilities concerning their surrounding PPR model

3 https://www.w3.org/RDF/.
4 https://www.w3.org/TR/rdf-schema/.
5 https://www.w3.org/OWL/.
6 https://www.w3.org/TR/rdf-sparql-query/.
7 https://www.w3.org/TR/shacl/.

elements and for the matching of semantic capability descriptions. A direct way of utilizing OWL is to model the notion of capability as an OWL class. One can then use OWL restrictions on OWL properties for representing properties and their constraints from the CSS Model. Applications can then introduce specific capabilities like *drilling* by means of sub-classing together with their relevant properties restricted in complex OWL class expressions. An example of such an expression in OWL Manchester syntax is "`Drilling and (depth some integer[<=15])`" to represent a capability for *drilling with a depth of max*. 15 mm. The specialization hierarchies for capability classes can be taken from standards such as DIN 8580 or VDI 2860, as proposed in [8]. This representational approach is similar to the one presented in [9] and extends it by equipping otherwise opaque capability classes with constraints on properties to account for their rich semantics.

Moreover, OWL reasoning can be utilized for capability matching, at least on a coarse level of detail. The conjunction formed from two capability class expressions—one offered by a resource and the other one requested for a process—can be checked for satisfiability by any standard OWL reasoner to test whether the two capabilities are compatible, meaning that their constraint sets can be jointly fulfilled. This technique goes back to the intersection matching proposed in [15]. As discussed in [16], issues with OWL's open-world assumption can be overcome by strictly controlling the ontological vocabulary used. This requires systematically including so-called closure axioms, which is presumably easy to achieve in a factory environment, e.g., disjointness between sibling capability classes from standard hierarchies. Still, this approach needs further research on potential issues with scalability and expressivity in comparison to alternative constraint solving methods when applied in real setups with complex capability descriptions on large property sets.

4.2 Executing skills using OPC UA

In recent years, the possibilities of implementing skills have been investigated and implemented in various publications and research projects, such as DEVEKOS, BaSys4.0/4.2, AKOMI or SmartMA-X [10, 17–22]. The vendor-independent communication standard Open Platform Communications Unified Architecture (OPC UA) has emerged as a promising approach for implementing skill interfaces. OPC UA has a high degree of diffusion in control technology due to its ability to provide a resource-neutral information model in its servers. The description of skills with all skill parameters can be mapped directly within this information model to enable unified control of resources. So far, standardized OPC UA information models (so-called "Companion Specifications"), focused primarily on data acquisition, e.g., for asset management or condition monitoring. These use cases require primarily *read-only* access which does not enable complete interoperability of machines in the sense of the Industrie 4.0 vision. Furthermore, a classic real-time capable communication standard for controlling the resources is still required. This currently means there can either be separate networks for data acquisition and control or a shared network that strongly limits the available traffic. Therefore, it is essential to enable *write* and, thus, control access to these machines and systems over OPC UA. Companion Specifications, such as the PackML state machine (OPC 30050) [23] or OPC UA programs (OPC 10000-10) [24], show first approaches for such access. However, there is a lack of a uniform and cross-domain concept for the realization of skills that OPC UA can provide.

As shown in Figure 1, a distinction between the skill interface and the actual skill is necessary. The skill interface can be defined as an OPC UA information model. Figure 2 shows a proposal for this model. A separate OPC UA ObjectType with the name *SkillType* is created, which provides the essential elements for the skill interface: The `name` of the skill is optional and serves as plain text for the user to quickly identify the skill. The `ontologyURL` must be specified and forms the reference to the capability in the ontology model (cf. Section 3.3). This reference can be used to identify the capability realized. The `skillStateMachine` represents a finite state machine similar to the aforementioned OPC UA for PackML or OPC UA programs. The `ParameterSet` contains necessary parameters to set or check during or after skill execution:

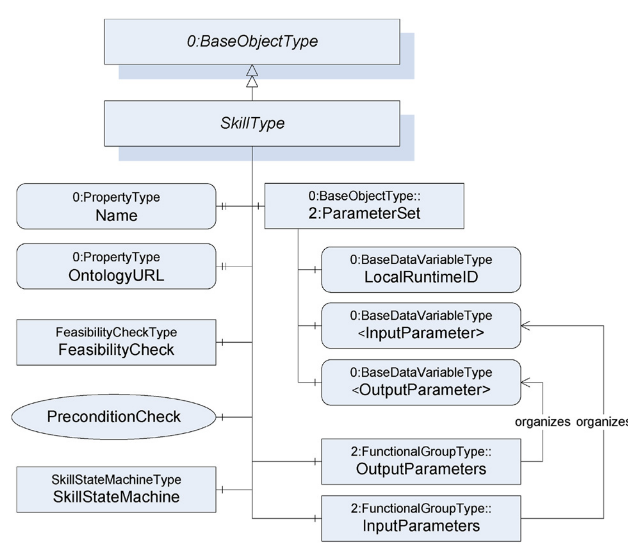

Figure 2: OPC UA skill metamodel.

– The `LocalRuntimeID` serves a client as a unique identifier to identify the skill for execution.
– The placeholder `<InputParameter>` is used to define any input parameters necessary for the execution or configuration of a skill. These can be, for example, position or speed parameters.
– The placeholder `<OutputParameter>` is used to define any output parameters that are returned by a skill. These can be, for example, current actual values (speed, velocity) during the skill execution, which are necessary for synchronization with other skills. Furthermore, it is also conceivable to create sensory skills, for example, for quality assurance, whose result is also represented by output parameters [19].

The optional `FeasibilityCheck` may be used to confirm the execution of complex skills in advance [20]. A feasibility check can also be implemented as a state machine in the OPC UA information model. To check the executability of a skill, the required input parameters are written into the parameter set of the feasibility check. The optional `PreconditionCheck` checks shortly before executing a skill whether the required resource fulfills all necessary conditions. This is especially needed if the execution depends on many other factors. In the assembly domain, this could be checking whether a required component is in stock, or, in the field of machine tools, checking if needed tools are available. Furthermore, the concept of skills can be integrated into OPC UA PubSub over Time-Sensitive Networking (TSN) to realize real-time capable communication for skills.

4.3 Describing services using the Asset Administration Shell

Only few publications, such as [11], distinguish the concept of service from capabilities and skills. Furthermore, in existing publications, there are no implementation examples for services. The exchange of information between companies must be based on mutually agreed semantics, which is standardized in the best case. Therefore, one solution to describe services is the Asset Administration Shell (AAS). The AAS is an Industrie 4.0 specification of a digital twin to enhance interoperability between systems of different vendors. Recently published specifications define a standardized AAS meta model and AAS interface [25]. The meta model specifies a set of elements used to create information models that are compliant with the concepts of Industrie 4.0, so-called Submodels (SMs), needed to represent several aspects and functionalities of a modeled asset. Using standardized SMs to ensure cross-company interoperability includes the standardization of information models for describing various aspects of modeled assets. The service is either demanded by the service requester or offered by the service Provider. Therefore, there must be two different SM templates.

The request contains the specification of required product or process requirements as well as the description of the required provision and may be part of the product AAS. The description of the provision, relevant for potential manufacturers, may include necessary certifications and a nondisclosure agreement if required. The different categories can be organized in Submodel Collections (SMCs) and can be described by properties. For instance, an SMC *TenderCritera* may represent all the information needed to find a suitable manufacturer based on, e.g., the required quantities, price specifications, CO_2 specifications, and delivery conditions. All SMCs and their properties are extended by a preset qualifier, indicating that the information is a requirement. Each property can also be described by a semantic ID. In case of a service, which is used between companies, standardized data elements are required. Therefore, repositories such as IEC Common Data Dictionary[8] or ECLASS[9] are recommended to ensure common semantics.

A service provider offers services that can be matched to required services. In this case, the SM service could be part of the factory or company AAS, based on a similar description as the requested service. The description is further extended by the capabilities which can provide the assured service to link the service and the corresponding capabilities.

5 Alternative approaches

Approaches to capabilities, skills, and services are a relatively new addition to comparable research approaches. Even before the terms of the CSS model formed, several approaches pursued similar goals, i.e., the encapsulation of automation functions and description of their interfaces. This section presents a differentiation of the CSS model with two comparable approaches.

5.1 Web Services and service-oriented architectures

Web Services are software systems that allow humans or machines to consume functionalities via a network.

8 https://cdd.iec.ch/cdd/iec61360/iec61360.nsf.
9 https://eclass.eu/eclass-standard/content-suche.

Interface descriptions in machine-readable formats are required for machine-to-machine interoperation [26].

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 [27]. Services are typically considered self-contained functions that may be composed of other services. They logically represent a recurring activity with a clearly defined input and output 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 [28].

The understanding of the term *service* as used in information technology differs significantly from the understanding expressed in this publication. 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 3.2).

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 [29]. Thus, services can be seen as an early precursor of skills according to the CSS reference model.

5.2 Module Type Package

A description of modular process units is provided with the Module Type Package (MTP). The MTP defines and describes data of the structure, information interfaces, process sequences, and functions (MTP services) of modules from automation technology. The combination and aggregation of components enables modular process units, known as Process Equipment Assembly (PEA). A PEA is developed once and contains the physical design of a process step to be implemented as well as the information technology interface to higher-level systems [30].

An MTP is a module description to allow module integration into a modular process plant. Modules provide MTP services with a predefined behavior (procedure) and a standardized interface that are offered externally with their description as MTP.

MTP concepts can be related to the aforementioned capabilities and skills, see Section 3.3. A description of an MTP service provides a specification of a function without invocation information and can thus be considered as a capability. A skill being an executable implementation may be compared to the implemented MTP service including procedures. Additional skill information, e.g., about the invocation interface and parameters are also included in an MTP—however, the State Machine is not explicitly modeled in an MTP.

Thereby, a PEA is the described resource of the aligned PPR approach at the CSS model, which provides skills. In contrast to the service of the CSS model, the complete MTP description is at capability and skill level and does not consider business, compliance, or commercial aspects.

Another relation to the concept of capability and module descriptions with its functions is given by the term "super-service" in [31]. A super-service is described as a conceptual planning artifact that contains the union of all process engineering services and procedures. The objective is to break down module boundaries and describe them in a manner comparable to capabilities, i.e., independent of modules or resources.

6 Conclusion and outlook

Flexible production is promising to meet the challenges of fluctuating markets. A current starting point for this is to enable an automatic comparison of the functions required for producing a product with the functions provided by machines and plants during production planning and operation. Such automatisms require that the components and stakeholders involved describe and communicate these functions with the same understanding on all levels of detail.

This paper proposes an abstract CSS reference model that defines the concepts capability, skill, and service and relates them to the concepts of the PPR approach. The CSS model reflects a notion of function on three levels of abstraction. It addresses questions about the set of interconnected function-related concepts and their relationships for flexible production. Furthermore, it combines the mostly isolated solution approaches described in the literature into a comprehensive reference model that enables interoperable solutions. Potential technical approaches for implementing the presented concepts were proposed.

However, the presented model is a first conceptual model that we created as a result of projects and cross-organizational working groups under the umbrella of *Plattform Industrie 4.0*. This article focused on the CSS model without looking in detail into industrial use cases. Readers can find a detailed analysis of potential use cases in [32]. Furthermore, there are still open challenges: At what level of granularity do capabilities, skills and services have to be modeled? Is there even one "correct" level of detail at all, or does it depend on the use case? And how does a CSS model scale when applied in real-world production? In addition, further research activities must be undertaken so that a comprehensive and consistent mapping of the model to selected technologies can be offered. A promising approach to use AASs has been started within the *Industrial Digital Twin Association* (IDTA).

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Research funding: None declared.
Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

[1] J. Pfrommer, M. Schleipen, and J. Beyerer, “PPRS: production skills and their relation to product, process, and resource,” in *18th IEEE ETFA 2013*, Cagliari, Italy, IEEE, 2013, pp. 1—4. September 10-13, 2013.
[2] R. Froschauer, A. Köcher, K. Meixner, S. Schmitt, and F. Spitzeret,, “Capabilities and skills in manufacturing: a survey over the last decade of ETFA,” in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2022, pp. 1—8.
[3] M. Schleipen, A. Lüder, O. Sauer, H. Flatt, and J. Jasperneite, “Requirements and concept for plug-and-work,” *at-Automatisierungstechnik*, vol. 63, no. 10, pp. 801—820, 2015.
[4] Verein Deutscher Ingenierure, *VDI/VDE 3682: Formalised Process Descriptions*, Berlin, Beuth Verlag, VDI/VDE, 2005.
[5] S. Borgo, E. M. Sanfilippo, and W. Terkaj, “Capabilities, capacities, and functionalities of resources in industrial engineering,” in *Proceedings of the Joint Ontology Workshops 2021 Episode VII: The Bolzano Summer of Knowledge co—located with the 12th International Conference on Formal Ontology in Information Systems (FOIS 2021), and the 12th International Conference on Biomedical Ontologies (ICBO 2021)*, E. M. Sanfilippo, O. Kutz, N. Troquard, T. Hahmann, C. Masolo, R. Hoehndorf, Eds., 2021.
[6] Plattform Industrie 4.0, *White Paper: Describing Capabilities of Industrie 4.0 Components*, Bundesministerium für Wirtschaft und Energie, 2020.
[7] Y. Liu, L. Wang, X. V. Wang, X. Xu, and P. Jiang, “Cloud manufacturing: key issues and future perspectives,” *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 9, pp. 858—874, 2019.
[8] A. Köcher, C. Hildebrandt, L. M. Vieira da Silva, and A. Fay, “A formal capability and skill model for use in plug and produce scenarios,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2020, pp. 1663—1670.
[9] M. Weser, J. Bock, S. Schmitt, A. Perzylo, and K. Evers, “An ontology-based metamodel for capability descriptions,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2020, pp. 1679—1686.
[10] K. Dorofeev and A. Zoitl, “skill-based engineering approach using OPC UA programs,” in *Proceedings IEEE 16th International Conference on Industrial Informatics (INDIAN)*, Piscataway, NJ, IEEE, 2018, pp. 1098—1103.
[11] A. Belyaev and C. Diedrich, “Erhöhung der Flexibilität und Robustheit zwischen Interaktionspartnern durch das Merkmalmodell,” *at — Automatisierungstechnik*, vol. 67, no. 3, pp. 193—207, 2018.
[12] DIN EN 62264-3, *Enterprise-Control System Integration: Part 3: Activity Models of Manufacturing Operations Management*, Berlin, Beuth Verlag, 2007.
[13] A. Vennesland, J. Cornelis de Man, P. Haro, E. Arica, and M. Oliveira, “Towards a semantic matchmaking algorithm for capacity exchange in manufacturing supply chains,” in *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 3, Setubal, Portugal, ScitePress, 2019, pp. 466—472.
[14] A. Köcher, C. Hildebrandt, B. Caesar, et al., “Automating the development of machine skills and their semantic description,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2020, pp. 1013—1018.
[15] L. Li and I. Horrocks, “A software framework for matchmaking based on semantic web technology,” in *WWW*, G. Hencsey, B. White, Y. -F. R Chen, L. Kovács, S. Lawrence, Eds., ACM, 2003, pp. 331—339. Available at: http://dblp.uni- trier.de/db/conf/www/www2003.html#LiH03.
[16] S. Grimm, “Semantic matchmaking with nonmonotonic description logics,” in *Studies on the Semantic Web*, vol. 1, Amsterdam, IOS Press, 2009.
[17] S. Jungbluth, J. Hermann, W. Motsch, et al., “Dynamic replanning using multi-agent systems and Asset Administration Shells,” in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Piscataway, NJ, IEEE, 2022, pp. 1—8.
[18] S. Malakuti, P. Zimmermann, J. Grothoff, et al., “Challenges in skill-based engineering of industrial automation systems,” in *2018 23nd IEEE International Conference on Emerging Technologies and Factory Automation*, IEEE Industrial Electronics Society. IEEE, 2018.
[19] V. Hammerstingl, “Steigerung der Rekonfigurations fähigkeit von Montageanlagen durch Cyber—Physische Feldgeräte,” Ph.D. thesis, 2020.
[20] M. Volkmann, A. Sidorenko, A. Wagner, J. Hermann, T. Legler, and M. Ruskowski, “Integration of a feasibility and context check into an OPC UA skill,” *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 276—281, 2021.
[21] M. Volkmann, T. Legler, A. Wagner, and M. Ruskowski, “A CAD feature-based manufacturing approach with OPC UA skills,” *Procedia Manuf.*, vol. 51, pp. 416—423, 2020.
[22] P. Zimmermann, E. Axmann, B. Brandenbourger, K. Dorofeev, A. Mankowski, and P. Zanini, “skill-based engineering and control on field-device-level with OPC UA,” in *IEEE ETFA 2019*, Piscataway, NJ, IEEE, 2019.
[23] OPC Foundation, *OPC 30050 — UA for PackML (OMAC)*, Scottsdale, AZ, OPC Foundation, 2020.
[24] OPC Foundation, *OPC UA Specification Part 10: Programs: Release 1.04*, Scottsdale, AZ, OPC Foundation, 2017.
[25] S. Bader, E. Barnstedt, H. Bedenbender, et al., *Details of the Asset Administration Shell — Part 1 The Exchange of Information between Partners in the Value Chain of Industrie 4.0 (Version 3.0RC02)*, 2022. Available at: https://www.plattform-i40.de/IP/Redaktion/\ignorespacesDE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html [accessed: Aug. 19, 2022].

[26] D. Booth, H. Haas, F. McCabe, et al., "Web services architecture. W3C note," 2004. Available at: https://www.w3.org/TR/ws-arch/.W3C.
[27] OASIS, *Reference Model for Service Oriented Architecture*, 2006, Available at: http://docs.oasis-open.org/soarm/%v1.0/soa-rm.pdf [accessed: Jul. 21, 2022].
[28] The Open Group, *The SOA Source Book*, 2009, Available at: http://www.opengroup.org/soa/source-book/intro/index.htm.
[29] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," *IEEE Trans. Industr. Inform.*, vol. 1, no. 1, pp. 62–70, 2005.
[30] VDI Verein Deutscher Ingenieure e.V., "VDI / VDE / NAMUR 2658-4: Automatisierungstechnisches Engineering modularer Anlagen in der Prozessindustrie: Allgemeines Konzept und Schnittstellen," Berlin, Beuth Verlag, 2019.
[31] M. Blumenstein, A. Stutz, and M. Maurmaier, "Rollenbasierte Entwurfsmethodik für modulare Anlagen auf Basis von sogenannten Super–Services," in *Automation 2019*, Düsseldorf, VDI Verlag, 2019, pp. 197–214.
[32] Plattform Industrie 4.0, *Discussion Paper: Information Model for Capabilities, skills & Services*, Bundesministerium für Wirtschaft und Energie, 2022.

Bionotes

Aljosha Köcher
Helmut-Schmidt-Universität, Hamburg, Germany
aljosha.koecher@hsu-hh.de

Aljosha Köcher is a research group leader at the Institute of Automation Technology of Helmut Schmidt University in Hamburg. His research focuses on semantic models for capabilities and skills together with methods needed to integrate such models in development processes. Aljosha is a co-organizer of the yearly special session on capabilities and skills at ETFA.

Alexander Belyaev
Otto von Guericke Universität, Magdeburg, Germany
alexander.belyaev@ovgu.de

Alexander Belyaev is a researcher at the Institute for Automation Engineering of Otto-von-Guericke University in Magdeburg. His research focuses on semantic models for capabilities and services in the context of industry 4.0 and decentralized cross company industry marketplaces. He is involved in various standardization committees and working groups on these topics.

Jesko Hermann
Technologie-Initiative SmartFactory KL e.V., Kaisers-lautern, Germany
jesko.hermann@smartfactory.de

Jesko Hermann is researcher group leader at SmartFactory Kaiserslautern. His research focus is the dynamic generation of supply chains in production networks based on services, capabilites and skills.

Jürgen Bock
Technische Hochschule Ingolstadt, Ingolstadt, Germany
juergen.bock@thi.de

Jürgen Bock is a researcher and group leader for AI application in innovative production and logistics systems at Technische Hochschule Ingolstadt. His research focus is in the area of knowledge representation and reasoning for production and logistics, and their combination with data-driven AI.

Kristof Meixner
Christian-Doppler-Laboratory-SQI, TU Wien, Vienna, Austria
kristof.meixner@tuwien.ac.at

Kristof Meixner is a researcher at the Christian-Doppler-Laboratory-SQI at TU Wien. His research interests include aspects of reuse and variability in capability- and skill-based production systems engineering. Kristof is a co-organizer of the yearly special session on capabilities and skills at ETFA.

Magnus Volkmann
TU Kaiserslautern, WSKL, Kaiserslautern, Germany
magnus.volkmann@mv.uni-kl.de

Magnus Volkmann is researcher at University of Kaiserslautern. His research focus is flexible manufacturing in the context of industry 4.0, pursuing the approach of capability- and skills-based production and investigating integration into marketplaces.

Michael Winter
RWTH Aachen University, Aachen, Germany
m.winter@plt.rwth-aachen.de

Michael Winter has been a researcher at the Chair of Information and Automation Systems for Process and Material Technology at RWTH Aachen University since 2021. His research focuses on solution strategies of process control tasks with knowledge-based systems.

Patrick Zimmermann
Fraunhofer-Institut für Gießerei-, Composite- und Verarbeitungstechnik IGCV, Augsburg, Germany
patrick.zimmermann@igcv.fraunhofer.de

Patrick Zimmermann has been a research associate at the Fraunhofer Institute for Casting, Composite and Processing Technology IGCV in Augsburg since 2018. His research focuses on simplifying the engineering process of automated assembly systems and creating information models for data analytics use cases in different domains.

Stephan Grimm
Siemens AG, Munich, Germany
stephan.grimm@siemens.com

Stephan Grimm studied Computer Science in Karlsruhe, Germany, where he also did his PhD in the area of Symbolic AI and logics-based Knowledge Representation and Reasoning. He is now a senior key expert researcher in the Techology department at Siemens in Munich. His research interests are in ontologies, automated reasoning and explainable AI primarily for industrial applications centered around Industry 4.0.

Christian Diedrich
Otto von Guericke
Uni- versität,Magdeburg, Germany
christian.diedrich@ovgu.de

Christian Diedrich is head of the Institute of Control Technology of the Otto-von-Guericke-University of Magdeburg. He has worked in many German and European research projects that cover the entire engineering life cycle of production systems. In addition to that, he is active in national and international standardisation activities.