Capabilities, Capacities, and Functionalities of Resources in Industrial Engineering

Stefano Borgo[1], Emilio M. Sanfilippo[1] and Walter Terkaj[2]

[1]ISTC-CNR Laboratory for Applied Ontology, via alla cascata 56/C, 38123, Povo, Trento, Italy

[2]Istituto di Sistemi e Tecnologie Industriali Intelligenti per il Manifatturiero Avanzato (STIIMA-CNR) Via A. Corti, 12, 20133, Milano, Italy

Abstract

Industrial engineers decide the production organization discussing possible processes and available resources. To reason on resources, ontologies for industrial engineering need to model their characteristics. Indeed, a resource available in the shop floor can be associated with a process plan only if the resource has the capability to satisfy the plan requirements. Existing ontologies miss to characterize and distinguish the concept of capability, even when it is explicitly modeled, from other notions like capacity and functionality. We aim at filling this gap by enriching our previous work on manufacturing resources with an explicit representation of capabilities, capacities, functionalities and other related notions.

Keywords

manufacturing resource, capability, capacity, functionality, planning, ontology

1. Introduction

The modeling of *capabilities* plays a prominent role in the ontological representation of manufacturing resources. For instance, capabilities can be useful to classify resources for cataloging purposes, and are used to assess which type of resource meets the requirements in planning scenarios and, consequently, to decide the assignment of a task to a physical resource in scheduling contexts [1]. However, capabilities are only scarcely characterized and distinguished from notions like *capacity* and *functionality*, these latter being used as well in resource modeling [2, 3]. To fill this gap, we present a proposal on how to conceive these notions within an ontological framework based on previous works on both resources modeling [4, 5] and functions (in the engineering sense) [6, 7, 8]. Since our primary purpose is to specify how resources capabilities, capacities, and functionalities can be understood, the paper takes a conceptual stance rather than a formal one. We will therefore give only some hints on the formal representation, which remains to be addressed as part of future work.

Before looking at our proposal, we report on some research results relevant for our study; readers can refer to [4, 9] for further references.

FOMI 2021: 11th International Workshop on Formal Ontologies meet Industry, held at JOWO 2021: Episode VII The Bolzano Summer of Knowledge, September 11–18, 2021, Bolzano, Italy

stefano.borgo@cnr.it (S. Borgo); emilio.sanfilippo@cnr.it (E. M. Sanfilippo); walter.terkaj@stiima.cnr.it (W. Terkaj)

Sarkar and Šormaz [3] identify four main occurrences for the term 'capability' in manufacturing with respect to: (i) *process capabilities*, i.e., the *goals* that manufacturing processes are meant to achieve, e.g., a hole feature as the goal of a drilling process; (ii) *machine-tool capability*, i.e., the *behavior* of either a whole machine or (some of) its components due to their mechanistic structures, e.g., the rotating behavior of a drill; (iii) *shop level capability*, i.e., the capability of an industrial shop floor resulting from the aggregation of the capabilities of the single resources in the shop floor; finally, (iv) *manufacturing capability*, i.e., what a company is able to produce. To model resources capabilities, the authors refer to capabilities generalizing (ii) and propose an interpretation which is not behavior-centric. In this view, a resource capability is a property *constraining* resources functionalities (see also [10]).[1] For instance, the turning function of a lathe is constrained by capabilities such as test bed size, spindle speed, head's position, etc. The notion of functionality is inherited from BFO [11]. Briefly, when applied to products, a functionality is a designed property that the product manifests during certain processes. Examples are the function of a hammer to drive in nails or the function of a heart pacemaker to regulate heart beating.

Järvenpää et al. [2] propose an ontology for representing resources capabilities called MaRCO. Among other things, the ontology allows (i) matching resources capabilities with plan requirements; (ii) aggregating *simple* capabilities into *combined* capabilities, e.g., *to pick* as the combined capability resulting from the aggregation of *to move* and *finger grasping*;[2] (iii) characterizing capabilities via *parameters*, e.g., the capability *to move* as being characterized by parameters relative to speed, acceleration, accuracy, workspace dimensions, etc. The authors consider capabilities as *functionalities* that resources are able to perform.

Solano et al. [12] discuss the use of ontologies for process planning, including how to conceive resources and their capabilities with an ontological approach. Building on the DOLCE ontology [13], the authors conceive capabilities as "the ability to carry out a type of activity" with a certain performance level. Similarly to [2], capabilities are described in terms of characterizing features and can be combined into complex capabilities. From this perspective, capabilities are represented as (DOLCE) individual qualities inhering in physical objects.[3] Examples of capabilities are *to load* (i.e., to be able to participate in a loading activity), *to cut*, *to measure* etc. In this framework, capacities are capabilities specifically represented with respect to amount of production. For example, the capacity of a grinding machine can be used to express its rate of producing machining flat surfaces with a certain level of roughness (see also [14]).

Finally, the standard ISO 15531 [15], informally known as MANDATE, conceives capability as the "quality of being able to perform a given activity". Capacity refers to the "capability of a [...] resource to perform its expected function from a quantitative point of view". An example of the latter is "the capacity of a [...] resource to produce a given quantity of output in a particular time period" (quotes from [15, p.5]).

To sum up, there is a general agreement in conceiving resources capabilities in tight connec-

[1]To be more precise, in Sarkar and Šormaz's view [3], a capability is a *disposition* in the sense of BFO [11], i.e., a property which objects bear and *possibly* manifest when certain conditions are met. For instance, the disposition of a magnet to attract objects made of metal manifests only when the latter objects are close enough to the magnet.

[2]Järvenpää et al. [2] do not provide a list of simple capabilities since this may depend on the specific production systems one works with.

[3]Recall that DOLCE individual qualities inhere in and depend on *single* individual entities.

tion to the type of process in which resources can participate to contribute to achieving the process goals. Disagreements however arise at the modeling level, especially with respect to the notion of functionality. In particular, differently from Solano et al. [12] where there is not an explicit link between capabilities and functionalities, capabilities *are* functionalities according to MANDATE [15] and Järvenpää et al. [2], whereas they *constrain* functionalities for Saarkar and Šormaz [3]. Surprisingly, there is almost no reference to the debate about functions [8, 16] although their evident role. The same for the notion of capacity that, even when introduced, is only marginally characterized. Finally, the terminological variety is a source of ambiguity and it does not help to compare the ontologies. For instance, what Saarkar and Šormaz [3] call *capabilities* correspond to *capability parameters* in Järvenpää et al.'s view [2].

We shall see in the next sections how capabilities, capacities, and functions can be integrated in a unified approach, which is however introduced only in a preliminary way. The paper is structured as follows. Section 2 provides a brief overview of previous work on resource modeling that is now further extended. In Section 3 we enter into the characterization of functions, whereas capabilities and capacities are analyzed in Section 4. Section 5 concludes the paper.

2. Resources and Activities

We introduce in this section the ontological characterization of manufacturing resources presented by Sanfilippo et al. [5]. This framework provides a basic modeling of notions like (manufacturing) activity, activity occurrence, state, and requirement, among others, which we will further enrich in the next sections to represent capabilities, capacities, and functionalities. Recall that Sanfilippo et al.'s work characterizes resources from a high-level perspective; that is, it tells what resources are from an ontological stance and what are the (minimal) conditions to qualify as such. However, it does not provide a classification of resources with respect to, e.g., vendors' catalogs, which can be plugged into the system (see [9] for an example of manufacturing resources classification with respect to both ontological and engineering criteria). Also, the proposed framework strives from the need of unifying multiple engineering views used in application areas such as manufacturing execution and control, detailed system design or production planning, early system design or strategic capacity planning. In this sense, the framework is meant to offer a flexible approach for disparate engineering scenarios.

According to [5], the key notion of manufacturing plan defines a set of manufacturing activities and requirements for their executions (see Sect. 3 for more information on activities). A manufacturing resource is a physical entity that satisfies some requirements specified by a manufacturing activity. To better understand this definition, let us introduce some other notions. From a formal perspective, we will use classical first-order logic (FOL) with function symbols; also, as common practice in knowledge representation, we omit universal quantifiers from the front-side of formulas when the scope of the quantifiers is clear.

We borrow the notions of activity and activity occurrence[4] from the Process Specification Language (PSL; ISO 18629) [17]. An occurrence is a happening in time, a.k.a. event in the literature: e.g. a particular machining event o_1 taking place in the manufacturing site s_1 at

[4]To simplify the terminology, we write 'occurrence' meaning 'activity occurrence'.

time t_1. Activities stand for repeatable patterns of behavior, i.e., event *types* that occurrences instantiate. Following PSL, we write $occOf(o, a)$ to mean that the activity occurrence o satisfies (instantiate) a. A manufacturing activity is a (possibly complex) transformation type whose occurrences take an input and transform it in the desired output; e.g., an activity establishing the conditions to give a certain shape to a plank (the input).

Definitions (D1)–(D3) introduce three relations which are used to define manufacturing activity in (D4) and manufacturing resource in (D5). We shall provide here only a general overview of the formulas; the reader can refer to [5] for more details.[5]

Most of the predicates appearing in the right-hand side of the definitions are primitives. Also, types exist in the quantification domain, hence they are represented through standard FOL predicates. The (primitive) relation $sat(x, y, t)$ tells that x *satisfies* (instantiates) type y at time t (see formulas below). $StateType$ is satisfied by states; $MfgActivityType$ by (manufacturing) activity occurrences; $MfgCapabilityType$ by physical endurants bearing certain capabilities (see Sect. 4), and $ItemType$ by either physical objects (e.g., a machine, tool, fixture, etc.) or amounts of matter (e.g., lubricant, amount of wood, glass, sand, etc.).[6] The binary predicate $hasReq(a, x)$ is the most general relation to bind an activity to a type which is a *requirement* for executing the activity. For instance, a machining activity may require a resource with the capability of cutting certain material types with specific parameters while satisfying given tolerances.

With these clarifications, definition (D1) states conditions that must hold to initiate the execution of a manufacturing activity. The formula tells that when an activity a has initial state requirement x, for each of its occurrences o, there is a state s satisfying x at the beginning of o.

D1 $$hasInitialStateReq(a, x) \equiv Activity(a) \wedge hasReq(a, x) \wedge StateType(x) \wedge \forall o(occOf(o, a) \rightarrow sat(o, x, beginOf(o)))$$

Similarly, definition (D2) tells that when an activity a has output x and final state y, each occurrence o of a ends with a physical item p satisfying x and in the state s satisfying y. The predicates PP and PC stands for *proper parthood* and *participation*, respectively (see [13]). $hasOutput(a, x, y)$ could be simplified to the binary $hasOutput(a, x)$ if reference to the final state is not required. We find useful the ternary variant of this relation because planners often specify items' states during production, e.g., cylinder head (in the state of being) held on pallet, valve (in the state of being) installed, etc.

D2 $$hasOutput(a, x, y) \equiv Activity(a) \wedge ItemType(x) \wedge StateType(y) \wedge PP(x, a) \wedge PP(y, a) \wedge \forall o(occOf(o, a) \rightarrow \exists p(sat(p, x, endOf(o)) \wedge sat(o, y, endOf(o)) \wedge PC(p, o, endOf(o))))$$

Definition (D3) tells that an activity has either item types or manufacturing capability types as resource requirements.

[5]The formalization in this paper revises, and sometimes improves, the formulas in [5]. In addition, we refer here to types instead of descriptions. Since types do not indeed depend on specific descriptions, the use of types simplifies the formalization. It follows that activities are types and that an activity can have a type as part.

[6]Differently from physical objects, amounts of matter are mereologically invariant physical endurants, that is, an amount of matter can not lose or acquire parts while keeping identity.

D3 $hasResourceReq(a, x) \equiv Activity(a) \wedge hasReq(a, x) \wedge (ItemType(x) \vee MfgCapabilityType(x))$

Definition (D4) states that a manufacturing activity is an activity that has initial state requirements (x), resource requirements (y), and expected outputs (z) in the final state (v).

D4 $MfgActivity(a) \equiv \exists xyzv(hasInitialStateReq(a, x) \wedge hasResourceReq(a, y) \wedge hasOutput(a, z, v))$

Finally, a manufacturing resource is a physical endurant that satisfies the resource requirement of some manufacturing activity.

D5 $MfgResource(r) \equiv PhysicalEndurant(r) \wedge \exists axt(MfgActivity(a) \wedge hasResourceReq(a, x) \wedge sat(r, x, t))$

As said above, the work in [5] was conceived to offer a general approach to bind resources to plans on the basis of the requirements specified in the latter. For instance, taking the use case presented in [5] and partially shown in Table 1, it is possible to state that a machine (e.g., *palletizing robot*) has a capability that matches a requirement (e.g., *loading on pallet*) needed to perform a certain activity (e.g., *op10*). Capabilities are however treated thereby only as "black-boxes" and no formal means are provided to match them to requirements. This approach can be useful in applications related to early system design or strategic capacity planning, but it is not sufficient when a more advanced technological analysis is needed to support manufacturing execution/control or detailed system design. In these cases, indeed, a feasible match between resources and activities can be (automatically) derived (like in [2]) only if resource capabilities are explicitly characterized.

Table 1
Excerpt of the use case data for powertrain valve assembly on a cylinder head (from [5]).

Activity ID	Activity Description	Input components	Activity requirements	Resources with capability
op10	Load cylinder head on pallet	Raw cylinder head	Loading on pallet	Palletizing robot
			Cylinder head workholding	Pallet
op20	Identify cylinder head	Raw cylinder head	Tool handling	Robot
			Identification	Vision system tool
			Cylinder head workholding	Pallet
op30	Apply sealant and lubricant	Raw cylinder head	Tool handling	Robot
			Sealing	Sealant dispensing tool
			Cylinder head workholding	Pallet
op40	Install intake and exhaust valves	WIP cylinder head, intake and exhaust valves	Tool handling	Robot
			Valve gripping	Gripper tool
			Cylinder head workholding	Pallet

We shall see in the next sections how to treat resource capabilities by extending what done in [5] in the light of the literature on engineering functions. This effort includes the analysis of short-term planning of manufacturing execution that asks to explicitly consider the capacity of resources to make a plan feasible. Therefore, the concept of capacity is needed to properly cover the whole range of intended applications. As we will see, we propose to distinguish capability and capacity since they provide different kind of information.

3. Engineering Functions and their Execution

In manufacturing one distinguishes process plans from production plans [5]. The first is the set of work instructions to convert a part from the initial to the required final form and may include the description of manufacturing processes, operational setup, process parameters, and possibly equipment and/or machine tool selection. The second determines which production resources are to be assigned to each activity occurrence on each workpiece, when each activity occurrence is to take place, while resolving contention for the resources [1]. In both cases, the underlying assumption is that a plan is a set of tasks. In the literature, the notion of task is modeled in very different ways. Within PSL we see them as types of activities.

More specifically, we take a task to be a pair of state types possibly with further constraints. The first state type constrains the state of the system for the task to be considered for realization (it constrains the so-called initial state). The second state type gives the task's goal, i.e., the expected state of the system after the task execution. For example, referring to *op40* in Table 1, we could have as initial state the presence of a cylinder head, valves and some tools for handling valves, and as goal the presence of a cylinder head with its installed valves. Further conditions might state that all valves must be installed in a specific position, the gripper must move along a specific direction to avoid damages, that no further tool or material is used, that the realization of the goal occurs within a certain time duration, etc.

Conceptually speaking, the comparison between the initial and final states in a task indicates a change which could (if realizable) be brought about via the realization of functions. For instance, the task identified by the presence of a cylinder head and valves as initial state, and the presence of a cylinder head with its installed valves as goal corresponds to a *join* function. Further conditions, like the participation of a gripper, may indicate how the change is expected to be performed. This simple idea, which guides our modeling approach, needs now to be refined.

We take a function to be a type of activity or, ontologically speaking, a class of events. For instance, the engineering function *joining a plate steel and a pipe by welding* collects the possible events whose starting point satisfies the initial state (the pre-conditions of the welding operation) and whose ending point satisfies the final state (the post-conditions of that welding operation). There can be ambiguities concerning the characterization of a function: functional talk by experts may not indicate to which object types (plate steel and pipe) the function (joining) applies nor the method (welding) with which to realize it. In a further example related to cutting wooden planks, the engineer may point to the need to have the plank (object) divided (function) in two pieces without worrying how this is achieved (method) perhaps because the tool to be used is already known, or perhaps because how to divide the plank is irrelevant for the plan.

Building on [18, 19, 6], we distinguish two notions of function. On the one hand, there are *ontological functions*, that is, types of events in which a certain change occurs but without specific constraints on how it is obtained or how long it takes. In the DOLCE ontology [13], ontological functions are classified as achievements. In this paper they are conceived as types of PSL activities in which pre- and post-conditions indicate a change of the ontological status of an entity and/or a variation/preservation of an ontological relationship (see Fig. 1). On the other

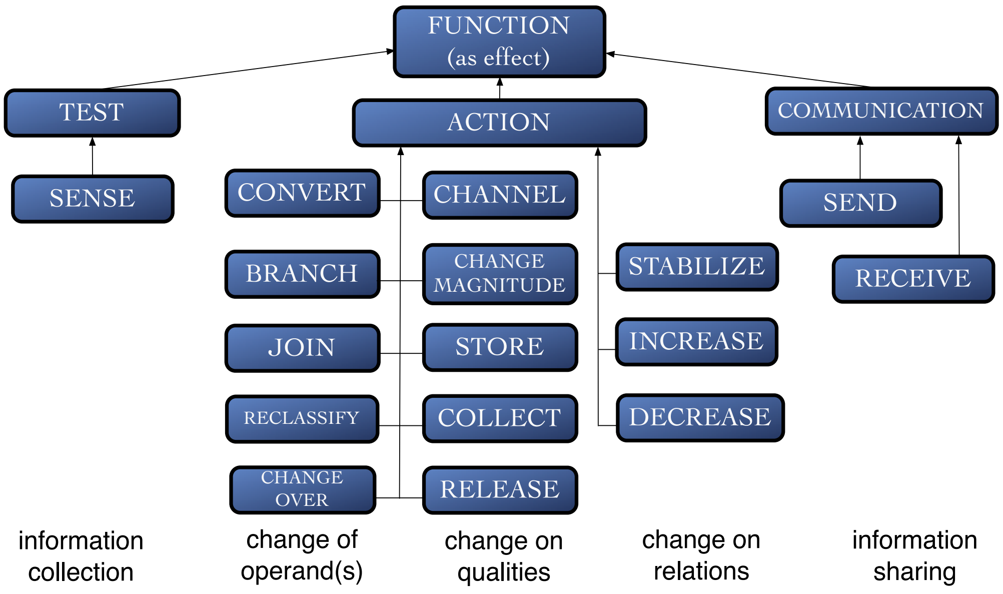

Figure 1: A (partial) ontological classification of engineering functions (from [18]).

hand, *engineering functions* are types of events in which the change given by initial and final states is required to satisfy further constraints among which, in particular, is the method with which the final state is achieved. These types of events are called accomplishment in DOLCE .

While a proper axiomatization is outside the scope of this paper, we can formally model the relationship between ontological and engineering functions as follows (here we assume that a classification of ontological functions is given, e.g., along the lines of the classification in Fig. 1, and that a list of engineering methods is fixed, e.g., as in the MTM methodology for the case of assembly tasks [20, 21]):

Ax1 $EngFnct(x) \rightarrow \exists yz(OntFnct(y) \land Method(z) \land sat(x, y) \land uses(x, z))$
(if x is an engineering function then there is an ontological function y that x satisfies, and a method z that x uses)

Then, recalling definition (D2), we have:

Ax2 $MfgActivity(x) \rightarrow \exists y(EngFunct(y) \land$
$\forall o(occOf(o, x) \rightarrow (sat(o, y, startOf(o)) \land sat(o, y, endOf(o)))))$
(if x is a manufacturing activity then there is an engineering function y such that every

occurrence of x satisfies the initial and final states of y)[7]

For instance, an ontological function might tell us that at the beginning there are some entities (e.g., a cylinder head, valves) and these entities change resulting in a single entity (e.g., a cylinder head with installed valves). Following Fig. 1, this description is a case of *join* function. When dealing with engineering functions, the same change is described stating the presence of entities (cylinder head, valves) at the start of the event, then stating that an agentive entity uses a gripper for some amount of time, in a certain way and according to some instructions, and that finally there is one entity (cylinder head with installed valves) when the event is completed. We will see in the next section how the conceptual difference between capabilities and capacities matches the above distinction between ontological and engineering functions.[8]

Finally, an engineering function execution is an activity occurrence in the PSL terminology (ontologically speaking, an accomplishment), in which the participating devices contribute to the achievement of the activity goal. A function execution is also known as function realization.

4. Capabilities and Capacities

Following the introduction of ontological and engineering functions, we now present capabilities and capacities as ontologically distinct notions. The purpose is to fill the gap, theoretical and practical, to enable engineers to generate, starting from a plan and a set of devices, the assignments for the plan execution. As said, the plan is seen as a sequence of tasks, and tasks are essentially sequences of functions. As mentioned, indeed, our previous work [4, 5] focused on concepts like plan, goal, and resource while taking for granted that the association between a plan and a resource is guaranteed by the plan itself. However, how this can be done in more specific terms remains challenging.

On the basis of what we discussed in the previous section, let us assume that a plan has been fixed and that it consists of a list of (perhaps only partially ordered) tasks, namely, engineering functions, with the characteristics that participants to these tasks must satisfy. We now face the following question: is a physical endurant (machine, tool, trained worker etc.) satisfying the characteristics required to be a participant of the plan realization?

In order to answer the question, we set the theoretical framework with the information needed to answer it. We propose to take *capability* to be the individual quality (in the sense of DOLCE) of an object that is associated with a function or with an activity. In particular, this means that a capability is a quality of an object relative to both an environment and an activity, i.e. a relational quality. Indeed, roughly speaking, what we have in mind corresponds to what one intends with expressions like ‘it has the capability to hold’ (said of a container), ‘it has the capability to carry’ (said of a truck), ‘it has the capability to cut’ (said of a blade). In these cases the quality is not of the object *per se* but of the object in a context (holding, carrying and cutting

[7]Note that there are two different sat predicates used in these formulas. The binary one in (Ax1) applies to pairs of types. The ternary sat in (Ax2) is temporalized: it applies to a triplet in which the first argument is an occurrent, the second a type and the third a time.

[8]Another important point in discussing functions is the level of granularity at which they are presented: functions can be simple (like transfer torque) or compound (like build a table). This is an important point that we leave for future work.

are implicitly limited to objects of a certain size, weight and material, even though at this level of representation the specific constraints might not be explicit). For this reason, a proposal could be to distinguish the *intrinsic* (individual) *qualities* of an entity (e.g. shape, weight and color), from the *relational* (individual) *qualities* of the same entity (e.g. capabilities to hold, to carry, to cut, to acquire information and so on). Future work on our proposal needs to consider how to treat this distinction in the light of the debate about relationships [22].

In industrial scenarios, most of these capabilities, and in particular those relevant to participation in a manufacturing activity, are *designed capabilities*. By designed capability, we mean the attribution, made by the designer and/or the producer, of a capability to the designed/produced object. The attribution of a designed capability to an object characterizes artifacts and requires conditions like the (implicit or explicit) presence of causal relationships relative to that capability in the object's design, and the successful result in a quality test [23]. The typical organization of work in industry ensures that among the capabilities of an object (i.e., the capabilities that it actually has at the time of consideration) there are its designed capabilities. Exceptions are assumed to occur in special cases known as failures (of devices) and misbehavior (of humans).[9]

An object that has a capability is an object whose interaction with the environment *causally contributes*[10] to the realization of an activity, that is, it can participate in the successful execution of that activity. The core idea is that, given an activity and the object's capabilities, the object can participate in an occurrence of that activity only if there is a role that fits its capabilities.

We have introduced capabilities as relational qualities relative to some functionality, i.e., an object that has a capability can contribute to realize some types of change. This determines that the object can play a role in an activity where that kind of transformation is needed. This, however, is not enough. A container has the capability to hold. Yet, in an activity occurrence the container has to contain a specific object for that particular activity to be realized. Similarly, a truck has to carry a specific type of object specifically required to be moved in order to successfully participate in a particular moving occurrence. In short, we need to move from being able to perform a function to being able to perform a function on certain objects and perhaps within specific constraints.

The notion of capacity is what we need here. Looking at how capacities are used in the literature (see Sect. 1), we propose to ontologically understand capacities as individual qualities that provide quantitative information. In this paper, we are interested in capacities related to capabilities as introduced above. The capacities of an object are its individual qualities that specify to which extent the object can realize a function, i.e., manifest a capability. Like capabilities, capacities are relational qualities that depend on the context in which the object is used. Examples of capacities are containment (volume available to perform a storage functionality depending on the type of entity, e.g. liquid, spheres, boxes and pallet) and throughput (e.g. output of parts relative to a production capability).

Without entering in details, we now show how this could be formally modeled. Recall that in our framework a capability is the quality of a physical endurant that can participate in an engineering function, and that a physical endurant that has a capacity has also at least a suitable capability:

[9]The problem may be in the plan itself because badly characterized or even physically unrealizable.
[10]The relation 'causally contribute' is here taken as primitive. It resembles the use in [24].

Ax3 $hasCapability(x, y) \to PhysicalEndurant(x) \land Quality(y) \land$
$$\exists zot(OntFnct(z) \land occOf(o, z) \land PC(x, o, t))$$
(if physical endurant x has capability y then there is an ontological function z and an occurrence of it in which x participates)

Ax4 $hasCapacity(x, y) \to PhysicalEndurant(x) \land Quality(y) \land \exists z(hasCapability(x, z))$
(if physical endurant x has capacity y then x has some capability z)

As in the formalization of functions in Sect 3, these axioms acquire full strength once a framework for the classification and interdependencies across capabilities and capacities is provided. Also, because resources only *possibly* participate in manufacturing occurrences, the use of modal logic would be more suited to represent modality. For instance, looking at (Ax3), it should be clear that x only possibly participates in o. Further work is therefore needed to coherently grasp these aspects in a robust formal way. Table 2 provides a summary overview for the terms discussed in the last two sections.

Table 2
Summary overview of terms

Term	**Preliminary characterization**	**Examples**
Ontological function	Type of event whose occurrences bring about a change in the application domain	To join, to collect, to cut, etc.
Engineering function	Type of event whose occurrences have to satisfy specific constraints in order to bring about a change in the application domain	To join a plate steel and pipe by welding, to cut steel with laser, etc.
Capability	A resource's quality relative to a function by which the resource can participate in the function realization and causally contribute to the achievement of the function final state	Having the capability to join, collect, cut, etc.
Capacity	A resource's quality specifying to which extent the object can realize a function	Containment (volume available to perform a storage function)

5. Conclusions

We proposed to separate the notions of capability and capacity, both seen as individual qualities of objects, and to use them to align objects with the functions they can perform. Also, the introduction of the distinction between ontological and engineering functions helps to reason in terms of transformations and ways of achievements. In this view, if a resource bears a capability, than it can participate in the execution of an activity with certain requirements. Our proposal is presented within the PSL approach following our earlier work, e.g., [5]. However, the conceptual approach we developed is general and can be reused in other frameworks.

Although the presented proposal has still a preliminary flavour, with respect to the state of the art, we explicitly spelled out how capabilities and functions can be conceived in a manner that is coherent with both applied ontology and industrial engineering. This is done by relying on the literature about engineering and ontological functions, which is seldom taken into account in existing works.

Further developments are needed to strengthen our proposal and bring it to the application level. A formal representation of the discussed notions is foremost required to explicitly capture their intended meaning. A deeper investigation of some ontological notions is also needed; e.g., we conceive capabilities and capacities as relational qualities. This view requires a precise treatment of what relational qualities are, as well as a representation of the corresponding *quality spaces* (i.e., abstract and commonly multi-dimensional spaces giving information about the values of qualities [13]). Finally. the approach needs an industrial benchmark to be tested against use cases.

References

[1] G. Chryssolouris, The Operation of Manufacturing Systems, Springer New York, New York, NY, 1992, pp. 321–413.

[2] E. Järvenpää, N. Siltala, O. Hylli, M. Lanz, The development of an ontology for describing the capabilities of manufacturing resources, Journal of Intelligent Manufacturing 30 (2019) 959–978.

[3] A. Sarkar, D. Šormaz, Ontology model for process level capabilities of manufacturing resources, Procedia Manufacturing 39 (2019) 1889–1898.

[4] E. M. Sanfilippo, W. Terkaj, S. Borgo, Resources in manufacturing, in: A. Barton, S. Seppälä, D. Porello (Eds.), Proceedings of the Joint Ontology Workshops 2019 (JOWO), volume 2518, CEUR Workshop Proceedings, 2019, pp. 1–12.

[5] E. M. Sanfilippo, W. Terkaj, S. Borgo, Ontological modeling of manufacturing resources, Applied Ontology 16 (2021) 87–109.

[6] R. Mizoguchi, Y. Kitamura, S. Borgo, A unifying definition for artifact and biological functions, Applied Ontology 11 (2016) 129–154.

[7] S. Borgo, M. Carrara, P. Garbacz, P. E. Vermaas, A formalization of functions as operations on flows, Journal of Computing and Information Science in Engineering 11 (2011) 031007 1–14.

[8] S. Borgo, R. Mizoguchi, B. Smith, Applied Ontology. Special Issue on the Ontology of Functions, Applied Ontology Journal 6 (2011) 99–163.

[9] E. M. Sanfilippo, S. Benavent, S. Borgo, N. Guarino, N. Troquard, F. Romero, P. Rosado, L. Solano, F. Belkadi, A. Bernard, Modeling manufacturing resources: An ontological approach, in: IFIP International Conference on Product Lifecycle Management, Springer, 2018, pp. 304–313.

[10] A. Sarkar, D. Šormaz, D. Koonce, S. Farah, Developing a resource-based manufacturing process capability ontology, in: D. A. Rossit, F. Tohmé, G. Mejía Delgadillo (Eds.), Production Research, Springer International Publishing, Cham, 2021, pp. 293–306.

[11] R. Arp, B. Smith, A. D. Spear, Building ontologies with Basic Formal Ontology, Mit Press, 2015.

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

[13] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, L. Schneider, DOLCE: A descriptive ontology for linguistic and cognitive engineering, WonderWeb, D18 (2003).

[14] L. Solano, P. Rosado, F. Romero, Knowledge representation for product and processes development planning in collaborative environments, International Journal of Computer Integrated Manufacturing 27 (2014) 787–801.

[15] ISO, ISO 15531-31: Industrial automation systems and integration - Industrial manufacturing management data - Part 31: Resource information model, 2004.

[16] M. S. Erden, H. Komoto, T. J. van Beek, V. D'Amelio, E. Echavarria, T. Tomiyama, A review of function modeling: Approaches and applications, Ai Edam 22 (2008) 147–169.

[17] M. Grüninger, Using the PSL ontology, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, 2009, pp. 423–443.

[18] S. Borgo, A. Cesta, A. Orlandini, A. Umbrico, Knowledge-based adaptive agents for manufacturing domains, Eng. Comput. 35 (2019) 755–779.

[19] R. Mizoguchi, Y. Kitamura, A functional ontology of artifacts, The Monist 92 (2009) 387–402.

[20] D. Almeida, J. Ferreira, Analysis of the methods time measurement (mtm) methodology through its application in manufacturing companies, in: 19th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2009), 2009.

[21] G. Fantoni, S. Al-Zubaidi, E. Coli, D. Mazzei, Automating the process of method-time-measurement, International Journal of Productivity and Performance Management 70 (2021) 958 – 982.

[22] N. Guarino, G. Guizzardi, "we need to discuss the relationship": Revisiting relationships as modeling constructs, in: International Conference on Advanced Information Systems Engineering, Springer, 2015, pp. 279–294.

[23] S. Borgo, M. Franssen, P. Garbacz, Y. Kitamura, R. Mizoguchi, P. E. Vermaas, Technical artifacts: An integrated perspective, Applied Ontology Journal 9 (2014) 217–235.

[24] S. Borgo, R. Mizoguchi, A first-order formalization of event, object, process and role in yamato, in: FOIS 2014, volume 267 of *FAIA*, IOS Press, 2014, pp. 79–92.