Formal Ontology in Information Systems
N. Aussenac-Gilles et al. (Eds.)

doi:10.3233/FAIA231134

# Manufacturing Resources, Capabilities, and Engineering Functions: Towards an Ontology-Based Integration

Francesco COMPAGNO [a,b,1], Stefano BORGO [a] and Emilio M. SANFILIPPO [a,c] and Walter TERKAJ [d]

[a] *CNR-ISTC Laboratory for Applied Ontology, Via alla cascata 56/c, 38123, Trento*
[b] *DISI, Università di Trento, Via Sommarive, 9, Trento, Italy*
[c] *CNR-ISTC, via Paolo Gaifami 18, 95126, Catania, Italy*
[d] *CNR-STIIMA, Via A. Corti, 12, 20133, Milano, Italy*

ORCiD ID: Francesco Compagno https://orcid.org/0000-0003-1002-608X, Stefano Borgo https://orcid.org/0000-0001-6001-2765, Emilio M. Sanfilippo https://orcid.org/0000-0003-2511-2853, Walter Terkaj https://orcid.org/0000-0001-8902-361X

**Abstract.** The representation of manufacturing resources plays a fundamental role in engineering modeling scenarios. These are characterized in a number of ways by taking into account their physical properties, capabilities, roles, etc. In this context, notions like capability, process, and functionality are used in different manners, hence it is not clear how different approaches can interoperate. The aim of this work is to propose an ontology for manufacturing that represents assets involved in manufacturing operations, their characteristics and relations, as defined in a list of requirements. This contribution stems from the existing literature and, in particular, integrates recent works related to the modeling of manufacturing resources, engineering functions, and capabilities. The ontology takes advantage of DOLCE as foundational ontological framework. The relevant axioms are presented and commented on with respect to the identified requirements.

**Keywords.** manufacturing, capabilities, functionality, resources, ontology integration, manufacturing systems design

## 1. Introduction

Researchers in applied ontology have been focusing on engineering notions in manufacturing for more than 20 years and considerable experience has been collected on how to model different areas of manufacturing, from production to functional modeling, from resource to system modeling [1].

The ontological study of manufacturing has attracted attention for different reasons, among which are the broad variety of modeling constraints that it presents and the economic impact to which it contributes. However, the domain is complex and legacy issues

[1] Corresponding Author: Francesco Compagno, francesco.compagno@unitn.it.

have slowed down the adaptation of industry's information systems to the ontological viewpoint. The result is that, as of today, a number of ontological models are available in essentially all the areas of manufacturing, but not much is known about how these models could be integrated. This integration, as it turns out, is a complex task. Attempts in this direction are carried out by different large initiatives like the European project (EU H2020) OntoCommons and the Industry Ontology Foundry (IOF).[2]

In this paper, we tackle the integration of ontological models in manufacturing planning. In particular, we focus on engineers' needs relative to *capability assessment*, a critical step in manufacturing planning where one needs to seamlessly reason on information as diverse as functionality, capacity, capability, plan, and resource. Our goal is not to develop new models. On the contrary, we aim to integrate material that has already been published and validated. This, we hope, will help to understand what is needed (and how) to move from the modeling of single engineering concepts to the modeling of broader engineering modeling frameworks. At the same time, however, in order to integrate existing ontologies, we sometimes need to adjust some notions to coherently fit them together under a common umbrella.

The next section introduces the target domain of our work and a guiding use case. Sect. 3 lists and motivates the requirements. Sect. 4 introduces the key concepts and how they have been modeled in the literature. Sect. 5 presents our unified theory building on the DOLCE ontology [2]. Sect. 6 shows how the resulting ontology satisfies the previously identified requirements. Finally, Sect. 7 concludes the paper.

## 2. Problem Statement and Use Case

Manufacturing system engineering aims to design and manage production systems, thus leading to the need of assessing if production systems are able to produce the intended products while being effective along various dimensions such as quality, cost, flexibility, delivery, and innovation [3]. Indeed, as goals and context evolve [4], one has to be ready to frequently (re)assess and, if needed, re-design the manufacturing capabilities of a production system without disrupting the model developed up to this point.

The design and management of production systems involve planning at different levels, thus asking to represent resources and their characterization in terms of both capabilities and the roles they play in operations (activities). Egbunike et al. [5], analyzing the existing literature, defined manufacturing capabilities as the "fundamental proficiencies that enable firms to achieve production related goals" that in turn should be aligned with the strategic goals [6]. The data related to a production system evolve during its lifecycle and this must be taken into account by planning activities that require data to be accessible, integrated, and distilled to match the evolving level of details of plans [7]. For instance, only limited details about products and manufacturing operations are available at the phases of early/conceptual system design or strategic capacity planning, whereas this information becomes more precise at the stages of detailed system design or production planning. Finally, during manufacturing execution and control, both the system configuration and process plans must be available. All these phases of the system lifecycle are strictly interconnected, and engineers need to access information at different levels of detail across all of them.

[2]For pointers and further information see: http://industryportal.enit.fr/, and https://ontocommons.eu/.

Following previous works [7,8], we adopt a use case to elicit (some of) the requirements for early system design and strategic capacity planning. These phases are selected because they represent well the complexity of manufacturing system engineering while posing relevant challenges, since it is needed to represent heterogeneous data with different levels of detail and possible incomplete information. In particular, we refer to the use case presented by Bruccoleri et al. [9], which deals with the design of a flexible manufacturing system for the production of mechanical parts. Relevant data can be grouped into three connected areas: product, process, and resource/system.

Product data is the set of products characterized by final and raw piece geometry, machining features, material, tolerance, bounding box size, and commercial data (e.g., cost, margin, orders, etc.). Machining features are related to corresponding machining operations that are characterized by required tool, cutting parameters (i.e., feed rate, cut speed, spindle speed), required power, type of operation, and precision level (e.g., roughing or finishing). Manufacturing system configurations consist of resources such as transporters, fixtures, tools, and machine tools that are needed to execute the type and number of operations to satisfy the demand for products. In particular, machine tools are characterized by the number and type of controlled axes (positive/negative travel, speed, acceleration), dimension of the working cube, spindle speed, nominal power, setup times, precision level (e.g., roughing or finishing), reliability, and investment cost. Fixtures are described by size and number of parts that can be clamped. The feasible assignment of a candidate resource to an operation depends on the requirements of the operation and the capability of the resources, for instance:

- Only a machine tool that is capable of performing finishing operations can execute an operation with the precision level of finishing.
- A machine tool can execute an operation only if it is capable of providing the required power and spindle speed.
- A fixture can be loaded on a machine tool only if its size fits the working cube of the machine.

## 3. Requirements for an Ontology of Manufacturing

From the domain analysis presented in the previous section, it follows that a conceptual model should accommodate several requirements:

*R.1* Definition of the basic notions involved in the characterization and use of manufacturing capabilities.
*R.2* Definition of relations among the basic notions relevant to describe and assess the manufacturing capabilities.
*R.3* Flexible modeling of capabilities to cope with different levels of detail.
*R.4* Support for automatic capability assessment, i.e., checking if a given set of assets can provide the required manufacturing capabilities to realize a plan.

More specifically, the list of basic notions (*R.1*) includes:

*R.1.1* Items and descriptions: engineers refer to both physical items (e.g., the artifact being worked now, the ongoing operation, the tool just attached to a machine) as well as descriptions, concepts, and classes to classify them.

*R.1.2* Process plan: the set of work instructions to convert an item from its initial form to the required final form [10]. The plan may include the description of manufacturing processes, operational setup, process parameters, etc., and may hold any detailed information on how these should be realized.

*R.1.3* Manufacturing resource: an asset participating in an engineering manufacturing process with a specific purpose (e.g., to start, control, or complete a type of transformation) and is roughly classified as a physical asset (e.g., machines), material (e.g., water, hydrogen), software (as special components of machines), or human.

*R.1.4* Manufacturing capability: an asset can play the assigned role of resource only if it has certain structural and functional capabilities.[3]

*R.1.5* Engineering function: in this context, engineering functions are activities (i.e., types of transformations), and are the building blocks of process plans.

*R.1.6* Engineering method: an engineering method is a specification of how an activity (transformation) should be realized.

*R.1.7* Condition: the realization of a process plan is a strictly controlled process that requires the verification that certain conditions are satisfied. Three types of conditions are particularly relevant in our context given that the control of the execution at runtime is not a target: preconditions (what should hold in the scenario before a transformation initiates), postconditions (what should hold in the scenario after the transformation is complete), and goals (what should hold at the end of the manufacturing process).

The above notions have been largely discussed in the engineering community and, more recently, have been investigated from an ontological perspective (see Sect. 4). Similarly, among the most relevant predicates and relations (*R.2*) there are:

*R.2.1* Representation of relations for the physical structure of whole products and their components (e.g., parthood, connection, etc.). [Competency question (CQ): what product must be manufactured? How is it structured?]

*R.2.2* Representation of relations for whole process plans and their parts (e.g., parthood, time ordering, etc.). [CQ: how is the product manufactured?]

*R.2.3* Representation of relations to link physical items to their descriptions (e.g., classification; cf. *R.1.1*). [CQ: what is needed to manufacture the product?]

*R.2.4* Representation of relations to make explicit the requirements of process plans in terms of input items, output items, and resource capabilities. [CQ: how are the activities executed and controlled?]

*R.2.5* Representation of relations to flexibly assign manufacturing resources to plans based on a feasible capability match. [CQ: which are the feasible options for production planning?]

*R.2.6* Representation of relations for the actual assignment of a manufacturing resource to an activity in a process plan (the *potential* assignment is considered in the previous point). [CQ: which production plan must be actually executed?]

*R.2.7* Satisfiability relation between a production plan and needed product volumes.. [CQ: will the selected production plan be able to meet the demand for products?]

[3] Regarding humans, one usually talks about competences and skills [11].

We do not list the specific requirements for *R.3* and *R.4,* which are left for future work (see, e.g., [12] for some further discussion). For the sake of this paper, we concentrate on *R.1* and *R.2* only, since the identification of classes and relations to be integrated is preliminary for other tasks.

As noticed above, the paper builds on top of a series of other works that have independently tackled these notions. The overall goal of this paper is to bring these results together into a unified ontological framework suitable for early system design and strategic capacity planning, which is seen as a further step toward a general modeling system. We insist that our purpose is to integrate these notions from the engineering perspective. Thus, as we will see, we present some assumptions justified within the engineering perspective like, e.g., the view that resources are objects with certain properties instead of roles played by those objects. We believe that these two views should be reconciled via the mapping that aligns the domain ontology (our target in this paper) with the foundational ontology one takes as reference.

## 4. The Key Concepts in the Literature

This section reviews the literature related to the key concepts defined in Sect. 3 (cf. *R.1*), in particular engineering functions (Sect.4.1), capabilities (Sect.4.2), and manufacturing resources (Sect.4.3). See the references of the works we present for additional material, including surveys and reviews.

### *4.1. Engineering Functions*

Modeling functions of components of engineering systems is hard and raises non-trivial questions about what engineering systems components are, thus leading to different views (see, e.g., [13,14]). For instance, some authors argue that, at least in the context of engineering systems, functions are externally grounded with respect to the functioning objects [15], while others (e.g., BFO's authors [16]) maintain opposite views.

Within this scope, several works were proposed to precisely explain what are the differences between different concepts of functions [17,18]. In particular, Compagno and Borgo [18] detail three different variants: 'engineering functions' are among those, which are functions where the method with which to carry the function out is specified, e.g. 'to join two metal tubes *by welding*'. We will elaborate on this further in the latter part of this section.

### *4.2. Works about Capabilities and Capacities*

The term *capability* is especially used in the context of manufacturing resource modeling, and is often accompanied by *capacity* [12,19]. The technical standard MANDATE [20] defines capability as "The quality of being able to perform a given activity. [...] defined by a group of characteristics that describes functional aspects of manufacturing resources or system[s]". It also states that "Capability is essentially a functional and qualitative concept", while capacity is a quantitative concept consisting in the "capability of a system, subsystem or resource to perform its expected function from a quantitative point of view". An example of capability for a machine tool could be "(the ability of) drilling a counterbore hole", while a capacity could be "the quantity of workpieces the machine can process in an hour".

The terms capability and capacity are broadly used in the literature with different, sometimes contrasting, meanings. For example, there is terminological overlapping between capabilities and functions, and also between capabilities and (manufacturing) processes, see e.g. the MaRCO ontology [12].

Given the lack of consensus and the importance of capabilities, several ontological analyses were carried out. In addition to the approach of the aforementioned MaRCO ontology, there is a line of research leveraging the top-level ontology BFO, and classifying capabilities as a subcategory of dispositions [21,22]. Another line of research suggests using a conceptual model which is aligned with the DOLCE top-level ontology [19,8,18]. In this approach, capabilities are considered as (DOLCE-)qualities, so that they are entities that must inhere in a bearer on whom they depend. Again, we will elaborate on this approach at the end of this section.

### 4.3. Works about Resource Modeling

Capabilities and capacities are often employed in resource modeling, since, in the context of manufacturing system design, one needs to match them against the requirements to manufacture products. In fact, capabilities and capacities should be modeled alongside process and production plans (see DolceEng [8]), as the first detail instructions about the operations needed to transform a workpiece into its designed form, while the latter assign suitable resources to the execution of the operations required by a process plan.

Work in this direction is presented in [7], where three different conceptualizations of manufacturing resources are compared and unified into a single model, where part of the terminology is taken from the Process Specification Language [23].

In conclusion, functionality, capabilities, resources, and related concepts are tightly coupled, but this connection is currently opaque in the literature. For this reason, this paper aims to propose a unifying conceptual model of such concepts. To do this, we shall use DOLCE as a guide, leveraging and extending previous works already individually aligned with DOLCE, as this makes the alignment easier. We especially consider the conceptualizations contained in three of the previously mentioned works, which, for the sake of clarity, we shall give mnemonic names to: Dolce+Functions for [18], Dolce+Resources for [7], and DolceEng for [8]. We shall briefly describe these three works, then, in the next section, we shall proceed with a tentative unification.

The main characteristic of Dolce+Functions is the distinction among three types of functions:

- **Systemic functions:** roles of technical artifact behaviors in causally contributing to the achievement of a predetermined goal of a given engineering system [17]. For example, 'close the vertical jaws of the spindle' could be the systemic function of a hydraulic cylinder in a machine tool.
- **Ontological functions:** the generalization of the engineering view of systemic functions into an ontological category that does not rely on the notion of system. Briefly, an ontological function is a transformation (ontologically, an event) that can be modeled in the language of the ontology and whose type is given by the comparison between its initial and final states [18]. The adjective 'ontological' is warranted due to the fact

that these functions are general, i.e., do not depend on domain or middle-level concepts, and are relative to the conceptual organization of the given upper ontology. For instance, assume that the upper ontology contains a category for quality, specialized in a location category. Then, 'change location' is an ontological function type, which is the type of the (systemic-)function 'Close the vertical jaws of the spindle'.

- **Engineering functions:** functions in which the change between the initial and final states has to satisfy further constraints, among which the method with which the final state is achieved [19,18]. E.g., 'to join two metal tubes *by welding*' is an example of an engineering function whose ontological function type is 'Join two separate objects'.

Dolce+Functions attempts also a definition of capabilities (and capacities), as do Borgo et al. [19] and DolceEng [8]. The general idea is to define capabilities as resources' qualities representing the ability to participate in the realization of a function and to contribute to the achievement of the final state of the function realization. On the other hand, capacities are defined to be resources' qualities that specify to which extent the resource can realize a function[4].

Finally, Dolce+Resources is the unified model presented in [7], which focuses on resources in the context of manufacturing activities. Briefly, one salient characteristic of this approach is that Manufacturing resources are defined as objects playing particular roles: an object is a manufacturing resource whenever it satisfies the requirements needed for the execution of a manufacturing activity. For example, a machine tool has the role of manufacturing resource when it complies with the specifics of a certain activity, e.g. it is able to drill a hole with a given depth, diameter, and precision in a given material.

## 5. Proposed Unified Ontology

To build our ontology, we use some predicates taken from the DOLCE ontology. We report them in Table 1 with brief examples. A more exhaustive description of DOLCE can be found in [2]. In addition, we rely on the extension of DOLCE presented in [25] (that we call 'Dolce+SocialRoles'), which introduces concepts (CN), their important subcategory of roles (RL), and descriptions (DS). These categories, which we subsume into the non-agentive social object category (NASO), are useful to model actual as well as hypothetical artifacts and events, that is, the entities engineers commonly talk about. For instance, one sometimes finds in the literature a terminological distinction between *as-designed-product* and *physical product* [26]; the latter is the physical object resulting from a manufacturing process, whereas the former is a more abstract entity "collecting" the properties that physical products are meant to satisfy (e.g., CAD models, bills of materials (BOM), plans).

Following Dolce+SocialRoles, concepts are properties that we reify into the domain of quantification and which classify other entities. Concepts are used in Dolce+SocialRoles [25] to model social roles and classify different persons at different times. Descriptions stand for the reification of concepts' definitions. For instance, the role of *President of the European Commission* is a concept defined by the description of the

[4]Notice that in this approach, capabilities, which are conceptualized as qualities, are analogous to the 'capacity functions' described in [24]: capacity functions are properties ascribed to devices, which indicate the capacity to perform a function.

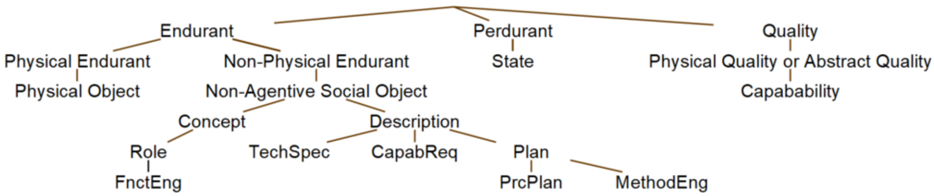


**Figure 1.** A taxonomy of the main classes of this paper, and some other DOLCE categories

*Constitutional Treaty*. Dolce+SocialRoles [25] binds concepts and definitions through the primitive relations *used by* (US) and its specialization *defined by* (DF), see (ax1) and (ax2), respectively. The former relation states that a description employs a concept (ax3), whereas the latter states that a concept is defined by a definition. In this view, a concept has a unique definition, which here we impose as an axiom (ax4), whereas multiple descriptions can use the same concept.

**ax1** $US(x,y) \rightarrow CN(x) \wedge DS(y)$
**ax2** $DF(x,y) \rightarrow US(x,y)$
**ax3** $DS(x) \rightarrow \exists y(US(y,x))$
**ax4** $CN(x) \rightarrow \exists! y(DF(x,y))$

In addition to the above predicates, we introduce some new ones accompanied by corresponding elucidations and collect them in Table 2. Then, we exploit the primitives of Table 2 to define further classes. In particular, we add predicates that correspond to the salient concepts mentioned in the requirements list of Section 3, such as process plan (requirement *R.1.2*) (df7), manufacturing resource (*R.1.3*) (df8), manufacturing capability (*R.1.4*) (df4), and conditions for manufacturing activities (*R.1.7*) (df5), (df6), (df3). The remaining requirements *R.1.5, R.1.6* are linked to the primitives `MethodEng` and `FnctEng`, while *R.1.1,* i.e., the duality between physical items and the classes of which they are instances, is satisfied exploiting DOLCE and the other primitives already introduced: physical endurants, descriptions, and concepts.

Figure 1 shows a taxonomy of the main classes introduced in this paper, together with (some of) the categories of DOLCE that they specialize.

Some of the following formulas are accompanied by a reference to indicate that the same or a similar formula was already presented in the literature: recall that Dolce+Resources stands for [7], DolceEng stands for [8], Dolce+Functions stands for [18]. Such formulas may have been modified in this work: for example, we replaced the terms 'artifact description', 'activity', 'proper-part-of' (PP), and 'satisfies'(sat) with 'technical specification', 'plan', 'constant-part-of'(CP), and 'classified-by' (CF), respectively, for sake of consistency and simplicity. Further differences, if any, are highlighted next to the single axioms. If no reference accompanies a formula, it means that it is original to this work. Throughout the axioms we use the (plan of the) assembly of powertrain valves on a cylinder head, an example recovered from [7], for the sake of illustration. We call this assembly plan 'valveAssemblyPlan#1', for the sake of brevity. The full example is also taken up again in Figure 2

**df1** $\mathtt{StateDescr}(x) \equiv \mathtt{DS}(x) \wedge \forall y,t(\mathtt{CF}(y,x,t) \rightarrow \mathtt{ST}(y))$ Dolce+Resources

| PRIMITIVE | DEFINITION |
|---|---|
| `PED`(x) | "*x* is a physical endurant". Concrete entities that are *wholly* present at each time they are present. For example, a car, a table, some quantity of water. |
| `ST`(x) | "*x* is a state". Differently than `PED`, States are not wholly present at each time they are present. Examples of states are 'being sat', 'having less than *p* pressure'. |
| `NASO`(x) | "*x* is a non-agentive social object". A (non-physical) object is a whole entity that depends on some community of agents. For example a currency or a law. |
| `PQ`(x) | "*x* is a physical quality". An individual property of a `PED`. E.g. the weight of a car. |
| `AQ`(x) | "*x* is an abstract quality". An individual property of a non-concrete entities. E.g. the reliability of some software. |
| `CP`(x,y) | "*x* is constant part of *y*". A parthood relation among `PED`, which lasts for the whole life of the containing entity, e.g. a car's door is constantly part of the car, if it never removed from it until the car is scrapped. |
| `I`(x,y,t) | "*x* inheres in *y* at time *t*". The (time-dependent) relation linking a quality to its bearer, e.g. the weight of a car *inheres* in that car. |

**Table 1.** List of DOLCE primitives used by this paper.

| PRIMITIVE | DEFINITION |
|---|---|
| `TechSpec`(x) | "*x* is a technical specification". A technical specification is a description that lists a series of technical properties and constraints that an entity (a material or immaterial - e.g., software - product, or a process) must fulfill up to some level. |
| `MethodEng`(x) | "*x* is an engineering method". An engineering method is a method recognized by the engineering community as suitable to achieve a goal. For example, welding, gluing, and fastening are engineering methods to achieve the joining of disconnected entities. |
| `FnctEng`(x) | "*x* is an engineering function". An engineering function is a (type of) event which brings about a desired goal through the application of some engineering method. |
| `Plan`(x) | "*x* is a plan". A plan is a description of a class of "similar" events, each counting as an execution of the plan, such that each event results in the achievement of a given goal. |
| `Capability`(x) | "*x* is a capability". Capabilities are qualities that inhere in entities able to carry out the corresponding function(s). |
| `hasReq`(x,y) | "*x* requires *y*". This is a relation restricted to descriptions. It says that executions of plan *x* need to satisfy description *y*. |
| `goalForAgent`(x,y) | "*x* is a goal for agent *y*". This relation makes explicit a goal that an agent has. |
| `occOf`(x,y) | "*x* is an occurrence of *y*". This relation links a plan to its executions (instances). This terminology is borrowed from PSL [23]. |
| `CF`(x,y,t) | "*x* is classified by *y* at time *t*". Alternatively "*x* satisfies *y* at time *t*". This is the temporalized relation that we use to link the reified classes to their instances. In [2] this relation holds between an endurant entity (e.g. a `PED` but not an event or state - otherwise it would be difficult to justify the temporal argument) and a concept; we extend the second argument to descriptions, too. |
| `actualResFor`(x,y,t) | "*x* is actually assigned to *y* at time *t*". This relation is used to indicate actual assignment of a resource to a plan, as opposed to the possible assignment that is introduced later, see (df8) |
| `StartOf`(x), `EndOf`(x) | These are logical functions that, given an occurrence, give its initial and final instant, respectively. |

**Table 2.** List of primitives introduced in this paper.

"State descriptions are descriptions that are satisfied by states only". Notice that originally this formula was weaker, consisting only of a left-to-right implication.

**df2** $\texttt{hasGoal}(x,y) \equiv \texttt{Plan}(x) \wedge \texttt{StateDescr}(y) \wedge \texttt{CP}(y,x) \wedge \exists z(\texttt{goalForAgent}(y,z)) \wedge \forall o \exists s(\texttt{occOf}(o,x) \rightarrow \texttt{CF}(s,y,\texttt{EndOf}(o)))$ Dolce+Resources

"Plan $x$ has goal $y$ if $x$ is an activity with state description $y$ as part and there is an agent $z$ having $y$ as goal, and for all occurrences $o$ of $x$ there is a state $s$ satisfying $y$ at the end of $o$". For example, a description of the powertrain valves assembled on the cylinder head is the goal of valveAssemblyPlan#1. Notice that in this formula agents are decoupled from processes. This is so since the agent may be unrelated to the manufacturing process (e.g., it may be a client requiring a product from a supplier), also different processes may be used to satisfy the same agent's goal.

**df3** $\texttt{hasCapabReq}(x,y) \equiv \texttt{hasReq}(x,y) \wedge \forall z,t(\texttt{CF}(z,y,t) \rightarrow \texttt{Capability}(z)) \wedge \forall o(\texttt{occOf}(o,x) \rightarrow \exists w,z,t(\texttt{PC}(w,o,t) \wedge \texttt{I}(z,w,t) \wedge \texttt{CF}(z,y,t)))$

"Plan $x$ has capability requirement $y$ if $x$ has requirement $y$ which classifies capabilities only, and each occurrence of $x$ is participated, at some time, by some entity bearing a capability classified by $y$". For example, one (of several) requirements for valveAssemblyPlan#1 is the capability of *loading* the raw cylinder head on a pallet.

**df4** $\texttt{CapabReq}(x) \equiv \exists y \texttt{hasCapabReq}(y,x)$

"$x$ is a capability requirement if there is a plan that has $x$ as capability requirement".

**df5** $\texttt{hasInputReq}(x,y) \equiv \texttt{hasReq}(x,y) \wedge \texttt{TechSpec}(y) \wedge \forall o \exists z(\texttt{occOf}(o,x) \rightarrow \texttt{CF}(z,y,\texttt{StartOf}(o)))$ DolceEng

"$x$ has input requirement $y$ if $x$ has requirement $y$, which is a technical specification, and for all occurrences $o$ of $x$, there is an object $z$ satisfying $y$ at the beginning of $o$". In our running example, a technical specification of the raw cylinder head is an input requirement of the plan. Notice that in the original reference `hasInputReq` is a primitive instead.

**df6** $\texttt{hasOutputReq}(x,y,z) \equiv \texttt{hasGoal}(x,z) \wedge \texttt{TechSpec}(y) \wedge \texttt{CP}(y,x) \wedge \forall o \exists s,p(\texttt{occOf}(o,x) \rightarrow \texttt{CF}(p,y,\texttt{EndOf}(o)) \wedge \texttt{CF}(s,z,\texttt{EndOf}(o)) \wedge \texttt{PC}(p,s,\texttt{EndOf}(o)))$ Dolce+Resources

"$x$ has intended output $y$ in $z$ means that $x$ has goal $z$ and $y$ is an technical specification that is part of $x$, and for every occurrence $o$ of $x$ there is an artifact $p$ and a state $s$ that are present at the end of $o$ and satisfy $y$ and $z$, respectively. Also, $p$ participates in $s$ at the end of $o$". The specification of the assembled cylinder head is the intended output of valveAssemblyPlan#1.

**df7** $\texttt{PrcPlan}(a) \equiv \exists y,z,w,v(\texttt{hasInputReq}(a,y) \wedge \texttt{hasCapabReq}(a,v) \wedge \texttt{hasOutputReq}(a,z,w))$ Dolce+Resources

"a process plan $a$ is a plan that has some input requirement $y$, some capability requirement $v$, and some output $z$ at some state $w$". The plan valveAssemblyPlan#1 is a process plan with respect to this definition. Notice that this formula is analogous to the definition of manufacturing activity in Dolce+Resources [7], only it uses predicates introduced in this work that were not originally present.

**df8** $\texttt{resourceFor}(r,a,t) \equiv \texttt{PED}(r) \wedge \texttt{PrcPlan}(a) \wedge \exists x,y(\texttt{hasCapabReq}(a,x) \wedge \texttt{CF}(y,x,t) \wedge \texttt{I}(y,r,t))$ Dolce+Resources

"$r$ is a resource for $a$ at $t$ means that $r$ is a physical endurant, $a$ a process plan with capability requirement $x$ and $r$ bears, at time $t$, some capability $y$ satisfying $x$ at time $t$". In our running example, a palletizing robot is a resource for the plan, since it has the loading capability. Note that, with respect to the original definition, the definient has been modified to highlight the central role of resources' capabilities.

**ax5** $\texttt{actualResFor}(x,y,t) \rightarrow \texttt{resourceFor}(x,y,t)$

"An actual resource for a process plan at some time is a resource for that plan at that very time". For example, a palletizing robot is an actual resource for the powertrain valve assembly plan when it has been decided to actually use that robot (and not, say, another robot or a manual operator).

This means that we distinguish the actual assignment to a plan (w.r.t. the possible assignment introduced in (df8)) by introducing a new, more specific, relation. Since we are subsuming the actual relation in the potential one, we are implicitly excluding the possibility of erroneous assignment related to the `resourceFor` relation.

Formula (ex1), taken from Dolce+Resources [7], would define manufacturing resources as a subclass of physical endurants. This is an alternative pattern to the conceptualisation of resources as roles, which correspond to a different and disjoint DOLCE-category (CN), but it can be reconciled if one accepts that (ex1) defines the *extension* of the role-concept, in place of its *intension*, which would be an individual concept. A definition of intensions of resources-role-concepts may, instead, look like (ex2). Nevertheless, we merely present (ex1) and (ex2) as possibilities consistent with the rest of the theory and commit to neither, since they have far-reaching consequences whose analysis we postpone to future work.

**ex1** $\texttt{MfgResource}(r) \equiv \exists a,t(\texttt{resourceFor}(r,a,t))$ Dolce+Resources

"$r$ is a manufacturing resource means that $r$ is a resource for process plan $a$ at some time $t$".

**ex2** $\texttt{MfgResource}(r) \equiv (\texttt{RL}(r) \wedge \exists r',t(\texttt{CF}(r',r,t)) \wedge \forall r',t(\texttt{CF}(r',r,t) \rightarrow \exists a(\texttt{resourceFor}(r',a,t))))$

"$r$ is a manufacturing resource means that $r$ is a role which classifies objects whenever they are resources for some process plan". Notice that this definition collects all *roles* that identify 'physical' resources, e.g. the $r$ in the definition could be a 'drill-resource-role', corresponding to the role of a 'physical'-drill when used as a manufacturing resource.

With this set of definitions at our disposal, we arrange the concepts of our theory in a taxonomy through the following axioms:

**ax6** $\texttt{Capability}(x) \rightarrow \texttt{PQ}(x) \vee \texttt{AQ}(x)$ DolceEng

"Capabilities are DOLCE physical or abstract qualities". (They are qualities of DOLCE physical endurants or of non-physical endurants, e.g., software). Note that in DolceEng [8] abstract qualities were not mentioned.

**ax7** $(\texttt{Plan}(x) \vee \texttt{TechSpec}(x) \vee \texttt{CapabReq}(x) \rightarrow \texttt{DS}(x)) \wedge (\texttt{DS}(x) \rightarrow \texttt{NASO}(x)) \wedge (\texttt{MethodEng}(x) \rightarrow \texttt{Plan}(x)) \wedge (\texttt{PrcPlan}(x) \rightarrow \texttt{Plan}(x)) \wedge (\texttt{FnctEng}(x) \rightarrow \texttt{CN}(x))$

"Plans, technical specification, and capability requirements are descriptions; engineering methods are plans; descriptions are non-agentive social objects; process plans are plans; engineering functions are concepts".

Lastly, we conclude this core theory by touching upon engineering functions and methods: axioms (ax8) and (ax9) link engineering functions and methods. For example, the function 'generating energy' can be achieved through different engineering methods, each requiring sub-tasks. E.g., if a solar panel is used, then some sub-tasks are converting solar energy to electrical energy, and altering electricity properties. Each sub-task is an engineering function that is implemented by another method. A function required by a method is *used-by* (US) the method (ax9). Generating energy is an ontological function, since it is independent of implementation, while 'generate-energy-with-solar-panels' is an engineering function *defined-by* (DF) the relative method, say 'employ-solar-panel' (ax8). Engineering functions are then similar to *qua-entities*: they are ontological functions *qua*-implemented-through-an-engineering-method. In Dolce+Functions [18], axioms (ax8) and (ax9) have a different shape but similar meaning to the simplified version expressed here using predicates from Dolce+SocialRoles [25].

Additionally, (ax10) says that engineering methods have goals, entailing, due to Axiom (ax8), that the same holds for (definitions of) engineering functions. Note that, differently than process plans, engineering methods need not fabricate products: e.g., transferring mechanical momentum through methods such as gears coupling or belt transmission does not produce a product. In this sense process plans may be interpreted as a specialization of engineering methods, though we do not commit to this view.

**ax8** $\texttt{FnctEng}(x) \rightarrow \exists y(\texttt{MethodEng}(y) \wedge \texttt{DF}(x,y))$ Dolce+Functions
"Each engineering function is defined by an engineering method".

**ax9** $\texttt{MethodEng}(x) \rightarrow \exists y(\texttt{FnctEng}(y) \wedge \texttt{US}(y,x))$ Dolce+Functions
"Each engineering method makes use of some engineering function".

**ax10** $\texttt{MethodEng}(x) \rightarrow \exists y(\texttt{hasGoal}(x,y))$
"Engineering methods have goals".

Axiom (ax10) concludes the formalisation. We also encoded the previous axioms in Common Logic and tptp, the resulting files can be accessed from GitHub[5].

The integration itself was performed by first isolating the primitive concepts of the considered works, then removing synonymous primitive concepts (or concepts that could be defined using the primitives of another work) by choosing a preferred terminology. Then, definitions of the main remaining concepts were attempted, and some of their properties were constrained using appropriate axioms. These last steps were carried out using a uniform methodology, thanks to the use of DOLCE as overarching ontology, e.g. by exploiting the relations of Dolce+SocialRoles (this entails significant differences with the original works, as it happened, e.g., for (ax8) and (ax9)). Several difficulties were encountered in this process and are mentioned in the conclusion.

## 6. Ontology Validation: First Steps

A basic requirement of any formal ontology is consistency. DOLCE is consistent [27], so we just need to check if the axioms of Section 5 do not cause contradictions. This is true because any model of DOLCE in which none of the categories of our ontology is instantiated is a model of our theory. This happens since we introduced two types of axioms: axioms of the form $A \rightarrow B$ (axioms (ax1) - (ax4), (ax6) and onwards), where $A$ is a proposition falsified when the categories of our ontology are empty, and definitions of the form $A \equiv B$, where the definient $B$ is false when the predicates of our theory are not instantiated.

[5] https://github.com/kataph/reconciled-manufacturing-ontology-serialisation

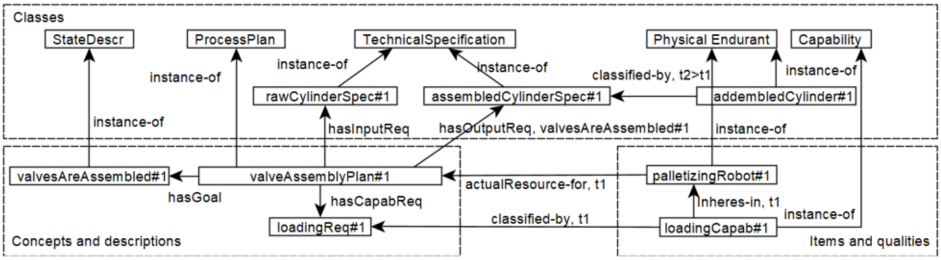


**Figure 2.** Part of a model of our theory. For sake of simplicity, we represent ternary relations as binary ones, with the third argument following the relation label ($t1$ and $t2$ are some time-regions, $t2 > t1$)

A more sought-after requirement is the consistency of the theory assuming that all its predicates are instantiated. This was verified using the macleod tool[6] for the fragment presented in this paper (without DOLCE axioms). Verifying the nontrivial consistency of the fragment together with DOLCE axioms is more difficult, but it is clear that the taxonomy of our theory is consistent with DOLCE taxonomy and, since no new axiom contains negations, the theory is not in contradiction with DOLCE. We also believe that the model represented in Figure 2 can be extended to a model of our theory and DOLCE, but this is quite complex to verify and more work is needed.

Another important requirement for ontologies is conceptual completeness, which can be understood either formally, as a property of a logical theory (i.e. a theory such that none of its categorical extensions contains concepts that are not definable using the primitives of the original theory [28]); or informally, as the request that the ontology should be able to express all the relevant concepts of the domain. It would be difficult to prove that our ontology is formally-conceptually-complete (and it likely is not), though we did proceed by listing numerous definitions from an initial set of primitives, as recommended by Bennet [28]. Regarding informal completeness, we argue that the requirements of Section 2 force our ontology to cover some relevant concepts of the domain, so we demand the informal check of completeness to the check of those requirements.

Coming to such requirements, we observe that Requirements *R.2.1* and *R.2.2,* which concern aggregation of entities, can be implemented by using appropriate mereologies. In particular, the decomposition of artifacts can be represented in terms of the temporalized parthood relation of DOLCE holding between physical endurants. We do not need to introduce new relations for this, unless one needs to specialize the parthood relation. The decomposition of plans can be modeled in a similar manner as we do, e.g., in (df2) and (df6). The treatment of parthood between descriptions requires however further analysis, e.g., in the light of the debate on slot mereologies [29].

Requirement *R.2.3* can be implemented by the classified-by relation holding between concrete individuals and descriptions (concepts), as briefly outlined in Section 5. Then, note that process plans can be associated with inputs, outputs, and resource capabilities by using the predicates defined in (df5), (df3), and (df6), respectively. This satisfies Requirement *R.2.4.*

Moreover, the predicates `actualResFor` and `resourceFor` allow us to speak of actual and feasible assignments, respectively (*R.2.5*, *R.2.6*). In our theory `resourceFor` is a defined relation that is satisfied depending on the matching between the capabilities

[6] `https://github.com/thahmann/macleod`

of an item and the capability requirements of a plan. Conversely, `actualResFor` is a primitive representing the assignment of a resource to a plan with which it is compatible.

Finally, requirement *R.2.7* is more complex and is not implemented here, we plan to discuss this aspect in future work.

## 7. Conclusions

In the paper, we address the integration of different research works on the modeling of manufacturing resources and functionalities, highlight some difficulties and provide a first module of an overarching theory with an analysis of the guiding requirements that it should satisfy. The integration plays an important role to reach a broader modeling of the engineering domain, and is based on a view that is today widely shared: (ontological, engineering) actual functions are (types of) transformations aimed to produce a desired outcome, whereas manufacturing resources are the concrete items used in manufacturing sites to execute occurrences of these transformations according to given requirements.

Although the integrated works share the same foundational framework (DOLCE), we have faced various challenges, some of which require further work – at both the foundational and domain levels. These include an in-depth analysis of descriptions (and concepts) in the scope of DOLCE-based ontologies. For instance, a technical specification commonly comprises the descriptions of different aspects of the item (process) at stake, e.g., components, dimensions, etc. How to make sense of these in a formal representation deserves further attention; a proposal has been presented in [30], but it should be further investigated how it can be adapted to our approach. Additionally, roles are treated in DOLCE as concepts classifying entities [2]. Also in Dolce+Resources [7] manufacturing resources are roles but they are formally represented as physical endurants satisfying certain conditions (ex1). This is a well-known manner to treat roles that is alternative to representing them 'directly' (say, e.g., as DOLCE-concepts (ex2)). The former manner may ease the formal representation (avoiding relying too much on reification mechanisms), while the latter may highlight the ontological characteristics of roles. Further analysis is needed to compare and reconcile these two (as well as others) approaches to roles.

From a domain perspective, notions like engineering method, ontological or engineering functionality are mostly based on the design literature, whereas they are now adapted to manufacturing contexts where one commonly speaks of plans, tasks, inputs, outputs, etc. The integration is challenging because of the diverse manners in which they are treated in the literature. In our view, for the sake of clarity, (engineering) functions – understood as types of transformations – are the building blocks of plans making explicit the requirements which resources have to satisfy to be used in executions.

Concerning the distinction between capabilities and functions, we have provided only a first core integration, which remains to be further developed in future work. Notice that, in our view, these two notions have to be kept apart. The difference between functionalities and capabilities lies indeed at the ground of the capability matching mechanism. Capabilities (and their descriptions at the level of requirements) allow to match resources to plans because of the functionalities of plans (e.g., a plan with a drilling-functionality requires a resource that is capable of drilling – with specific parameters).

## Acknowledgement

The work presented in this paper is partially funded by the European project OntoCommons (GA 958371).

## References

[1] Sanfilippo E, Robert I, et al. Formal ontologies in manufacturing. Applied Ontology. 2019;14(2):119-25.
[2] Borgo et al S. DOLCE: A descriptive ontology for linguistic and cognitive engineering. Applied Ontology. 2022 March;17(1):45-69.
[3] Spanos YE, Voudouris I. Antecedents and trajectories of AMT adoption: The case of Greek manufacturing SMEs. Research Policy. 2009;38(1):144-55.
[4] Tolio T, Ceglarek D, ElMaraghy HA, Fischer A, Hu SJ, Laperrière L, et al. SPECIES—Co-evolution of products, processes and production systems. CIRP Annals. 2010;59(2):672-93.
[5] Egbunike O, Purvis L, Naim MM. A systematic review of research into the management of manufacturing capabilities. Production Planning & Control. 2018;29(16):1349-66.
[6] Taps SB, Steger-Jensen K. Aligning supply chain design with manufacturing strategies in developing regions. Production Planning & Control. 2007;18(6):475-86.
[7] Sanfilippo EM, Terkaj W, Borgo S. Ontological modeling of manufacturing resources. Applied Ontology. 2021;16(1):87-109.
[8] Terkaj W, Borgo S, Sanfilippo EM. Ontology for Industrial Engineering: A DOLCE Compliant Approach. vol. 3240 of CEUR Workshop Proceedings; 2022. .
[9] Bruccoleri M, Capello C, Costa A, Nucci F, Terkaj W, Valente A. In: Tolio T, editor. Testing. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009. p. 239-93.
[10] Kamrani AK, Vijayan A. In: Kamrani AK, Nasr ESA, editors. Template-Based Integrated Design: A Case Study. Boston, MA: Springer US; 2008. p. 19-41.
[11] Arena D, Ameri F, Kiritsis D. Skill modelling for digital factories. In: IFIP International Conference on Advances in Production Management Systems. Springer; 2018. p. 318-26.
[12] Järvenpää E, Siltala N, Hylli O, Lanz M. The Development of an Ontology for Describing the Capabilities of Manufacturing Resources. Journal of Intelligent Manufacturing. 2019 Feb;30(2):959-78.
[13] Guarino N. Artefactual Systems, Missing Components and Replaceability. In: Artefact Kinds. Springer International Publishing; 2014. p. 191-206.
[14] West M. Developing High Quality Data Models. Burlington, MA: Morgan Kaufmann; 2011.
[15] Röhl J, Jansen L. Why Functions Are Not Special Dispositions: An Improved Classification of Realizables for Top-Level Ontologies. Journal of Biomedical Semantics. 2014;5(1):27.
[16] Spear et al D. Functions in Basic Formal Ontology. Applied Ontology. 2016;11.
[17] Mizoguchi R, Kitamura Y, Borgo S. A Unifying Definition for Artifact and Biological Functions. Applied Ontology. 2016 Jun;11(2):129-54.
[18] Compagno F, Borgo S. Towards a Formal Ontology of Engineering Functions, Behaviours, and Capabilities [Accepted for Publication to the Semantic Web Journal]; 2022.
[19] Borgo S, Sanfilippo EM, Terkaj W. Capabilities, Capacities, and Functionalities of Resources in Industrial Engineering. vol. 2969 of CEUR Workshop Proceedings; 2021. .
[20] ISO. ISO 15531-31: Industrial Automation Systems and Integration - Industrial Manufacturing Management Data - Part 31: Resource Information Model; 2004.
[21] Sarkar A, Šormaz D. Ontology Model for Process Level Capabilities of Manufacturing Resources. Procedia Manufacturing. 2019;39:1889-98.
[22] Merrell E, et al. Capabilities;. Available from: https://philpapers.org/rec/MERC-14.
[23] Grüninger M. In: Staab S, Studer R, editors. Using the PSL ontology. Springer; 2009. p. 423-43.
[24] Kitamura Y, Mizoguchi R. Characterizing Functions Based on Phase- and Evolution-Oriented Models. Applied Ontology. 2013;8(2):73-94.
[25] Masolo C, Vieu L, Bottazzi E, Catenacci C, Ferrario R, Gangemi A, et al. Social Roles and their Descriptions. In: KR; 2004. p. 267-77.
[26] Matsokis A, Kiritsis D. An ontology-based approach for Product Lifecycle Management. Computers in industry. 2010;61(8):787-97.

[27] Kutz O, Mossakowski T. A modular consistency proof for DOLCE. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 25; 2011. p. 227-34.

[28] Bennett B. The role of definitions in construction and analysis of formal ontologies. In: Logical Formalization of Commonsense Reasoning (2003 AAAI Symposium). AAAI Press; 2003. p. 27-35.

[29] Barton A, Toyoshima F, Vieu L, et al. The Mereological Structure of Informational Entities. In: FOIS; 2020. p. 201-15.

[30] Masolo C, Sanfilippo EM. Technical artefact theories: A comparative study and a new empirical approach. Review of Philosophy and Psychology. 2020;11(4):831-58.