Applied Ontology 16 (2021) 87–109
DOI 10.3233/AO-210242
IOS Press

Ontological modeling of manufacturing resources

Emilio M. Sanfilippo [a,*], Walter Terkaj [b] and Stefano Borgo [a]
[a] *Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy*
[b] *Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, STIIMA-CNR, Milano, Italy*

Abstract. Standards and ontologies for manufacturing understand resources differently. Because of this heterogeneity, misunderstandings arise concerning the basic features that characterize them. The purpose of the paper is to investigate how to ontologically model resources with the goal of facilitating the development of knowledge representation models for manufacturing. By reviewing the literature, we discuss and compare three approaches for the representation of resources depending on whether they are conceived in connection to either processes, plans or goals. By addressing the advantages and shortcomings of each view, we present a unifying perspective to enable the modeling of resources in an integrated manner. In this way, the intended meanings of the used notions are harmonized and, as a result, one can facilitate multiple experts to interact e.g., via data sharing and/or data integration procedures. Differently, by keeping three separated views, there is no guarantee that data coming from different parties will share common meanings even if the same terms are used. By the end of the paper, we present a case study to show the application of our approach and to compare it with an existing ontology for manufacturing.

Keywords: Manufacturing resource, planning, systems design

Accepted by: Bob Young

1. Introduction

The concept of resource is fundamental in various domains. Being a general term, it has been used in a variety of communities without becoming the subject of specific definitions, thus relying on the implicitly shared viewpoint among the community's members. This lack of characterization and the possible mismatches in the understanding of the term jeopardize efforts to share data models as well as to integrate information systems and services. For instance, according to the Oxford Dictionary of English (ODE), a resource is "[a] stock or supply of money, materials, staff, and other assets that can be drawn on by a person or organization in order to function effectively [...]." In this view, a resource is like an asset, i.e., an item that has value for and is owned by an agent. A resource does not need to be related to any action; what is required is that it has a value for some agent to "function effectively." For example, a wrench is of value to a mechanic for performing a job even though he does not use it.

Once we move from broad dictionary definitions to application scenarios such as manufacturing, the intended meaning of resource-related notions narrows to match engineering views. Despite this, there is little agreement among experts and stakeholders on how to conceive *manufacturing resources* (Sanfilippo et al., 2018; Sarkar and Šormaz, 2019). Consider the following definitions taken from industrial

*Corresponding author: Emilio M. Sanfilippo, Laboratory for Applied Ontology, ISTC-CNR, via alla cascata 56/C, Povo, 38123, Italy. E-mail: emilio.sanfilippo@cnr.it.

standards: "Any device, tool and means, except raw material and final product components, at the disposal of the enterprise to produce goods and services" (ISO 15531-1; ISO, 2004a); "A resource is an entity that provides some or all of the capabilities required by the execution of the enterprise activities and/or business processes. The types of resources involved in manufacturing operational management are: personnel, material, equipment (role based and physical asset) and process segments" (IEC 62264; IEC, 2013); "Means used by an activity to transform input into output" (ISO 20534; ISO, 2018); "[T]he costs, schedule, and other impacts from the use of a thing in a process" (ISO 16739; Liebich et al., 2013).

To comment on these views, in the case of ISO 15531-1 and ISO 20534, resources are things used to produce goods (or services), i.e., to transforms inputs into outputs. ISO 16739 and IEC 62264 are more general and do not explicitly refer to input-output transformations. In addition, differently from ISO 15531-1 and ISO 20534, they include materials in the range of resources. In most cases, the standards connect resources to the needs of activities or processes. ISO 15531-1 does not consider this connection mandatory and takes *availability* as the characterizing property. An information system based on this standard may not reliably interoperate with a system based on, e.g., IEC 62264 because of the mismatches in the employed models.

The purpose of this paper, which develops and extends the work presented by Sanfilippo et al. (2019), is to shed light on the modeling of manufacturing resources to foster the development of a reference computational ontology in the manufacturing domain (and possibly beyond). In particular, we review in Section 2 three approaches for resource modeling by addressing their advantages and drawbacks. Section 3 presents a way to integrate the main elements of these three approaches for a unifying treatment of notions like resource, plan, and goal, among others. Section 4 presents an example for resource representation in the context of manufacturing system design. The case study is used in Section 5 to compare our approach with an existing ontology for manufacturing, therefore, to show the potential applicability of our study for knowledge and data representation. Section 6 concludes the paper by addressing the need for future research work. Finally, the Appendix reports the (logical) predicates used in the paper.

2. Overview of approaches to manufacturing resources

We describe in this section three approaches emerging from the literature for the representation of manufacturing resources. Our purpose is to distinguish and identify different ways in which resources are understood. We use first-order logic (FOL) to highlight the differences between the approaches, although a complete axiomatization of each view is beyond our purposes. We also rely on the Unified Modeling Language (UML) Class Diagram notation to provide an intuitive, graphical understanding of the analyzed views. However, considering the differences between FOL and UML (Berardi et al., 2005), we will not attempt to provide an exact match between FOL axioms and UML diagrams.

As anticipated in the introduction, the terminology about manufacturing resources is ambiguous because experts and stakeholders (and, as a consequence, standards) rely on different and not well aligned vocabularies (Sarkar and Šormaz, 2019; Järvenpää et al., 2019). For the sake of clarity, we use the terminology of the Process Specification Language (PSL, ISO 18629; Grüninger, 2009b), since it provides the distinction between *activities* and *activity occurrences* that is important in discussing resources. The latter are *processes* occurring in time, whereas the former are the activity occurrences' descriptions, namely, their plans. The choice of interpreting PSL activities as plans is debatable because they can be also understood as *activity types* (Solano et al., 2014). However, the correspondence between PSL activities and plans is documented in the literature (Grüninger, 2009b). We will therefore use the terms activity and activity occurrence (or process) when discussing about ontological modeling choices.

Before moving to the next sections, we introduce some basic predicates that are common across the approaches we will discuss. As a remark on the notation, we write n-ary predicates (with $n > 1$) starting with a lowercase character, differently from unary predicates that start in uppercase. Exceptions are n-ary predicates taken from the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE; Masolo et al., 2003), which – following the original source – are all in uppercase.

Axiom (Ax1) introduces the relation of *participation* (PC) between endurants (ED) and perdurants (PD). Recall that the former are entities primarily existing in space and which are able to change at some extent without loosing their identities. Examples are amounts of matter, engineering tools, persons or features (holes, pockets, bumps, etc.), among others. The latter are entities happening in time (T) like cutting, welding, waiting and similar activity occurrences and states. The predicates ED, PD, PC, and T are inherited from DOLCE and they are commonly characterized across multiple ontologies, including PSL.[1] Axiom (Ax2) introduces the predicate for *presence* (PRE), also inherited from the DOLCE ontology. It holds between an entity and a period (or instant) of time during which it exists; $PRE(x, t)$ is read as *x is present at time t*. Definition (Def1) introduces the predicate for physical endurants (PED) which is inherited from DOLCE and is defined in this paper using our notation. Looking at the formula, the relation $in(x, r, t)$ is used to refer to the space region r where a physical endurant is located at time t. For the sake of the paper, we assume that the structure of space regions is minimally characterized, e.g., by the axioms of classical extensional mereology (Casati et al., 1999).[2] Axiom (Ax3) states that the predicate *sat* (for *satisfies*) is a primitive relation holding between an endurant (or perdurant) and a description it complies with. We use *Descr* as a primitive predicate that classifies descriptions like, e.g., the engineering specification of a device. Descriptions form a class which is disjoint with perdurants and physical endurants. We also assume that if descriptions have parts, these are descriptions, too. Axioms (Ax4) and (Ax5) introduce the primitive notions of *activity* and *activity occurrence*, respectively. The relation holding between an activity and its corresponding occurrence, i.e., $occOf(o, a)$ – read as *occurence o satisfies activity a* – is inherited from the PSL Core, see (Ax6).[3]

Ax1 $PC(x, y, t) \rightarrow ED(x) \wedge PD(y) \wedge T(t)$
(if x participates in y at t, then x is an endurant, y a perdurant, and t a time)

Ax2 $PRE(x, t) \rightarrow (ED(x) \vee PD(x)) \wedge T(t)$
(if x is present at t, then x is an endurant or a perdurant, and t a time)

Def1 $PED(x) \triangleq ED(x) \wedge \forall t\, \exists r (PRE(x, t) \rightarrow SpaceRegion(r) \wedge in(x, r, t))$
(x is a physical endurant means that x is an endurant such that whenever it is present at time t, it is located in a space region r)

Ax3 $sat(x, y, t) \rightarrow PRE(x, t) \wedge Descr(y)$
(if x satisfies y at time t, then x is present at time t and y is a description)

Ax4 $Activity(a) \rightarrow Descr(a)$
(if a is an activity, then a is a description)

Ax5 $ActOcc(o) \rightarrow PD(o)$
(if o is an activity occurrence, then o is a perdurant)

[1] The relation of *participation* is inherited from DOLCE rather than from PSL because it is more generally characterized in DOLCE with respect to perdurants rather than to specific perdurants classes such as activity occurrences.

[2] The representation of space can be further characterized in terms of, e.g., topological or geometric theories.

[3] In PSL Core, this predicate is written *occurence_of*. The reader can refer to the COLORE repository (Grüninger, n.d.) for a complete overview on PSL axioms.

Ax6 $occOf(o, a) \rightarrow ActOcc(o) \wedge Activity(a)$
(if o is an occurrence of a, then o is an activity occurrence and a is an activity)

The previous axioms hold in general while definition (Def2), given below, is specific to the manufacturing domain. We introduce it here because it provides a broad characterization of manufacturing activity occurrences and is not bound to any specific view we discuss later in the paper. According to (Def2), manufacturing activity occurrences (*MfgActOcc*) are activity occurrences such that at their completion a new product (*Product*) starts to exist. This definition embeds the view that the aim of a manufacturing process is to *fabricate* products. In this sense, procedures like cleaning or tuning a device are seen as parts of manufacturing activity occurrences but they are not manufacturing activity occurrences themselves (see footnote 4). The view is contentious, yet well documented in the literature (Alting, 1982; Colledani et al., 2008). Note also the constraint that a new product starts to exist at the end of the activity occurrence. This is a simplifying assumption that we keep throughout the paper. Admittedly, a product may start to exist at any stage of a manufacturing process depending on the modeling assumptions (Masolo and Sanfilippo, 2020). Also, we assume that a product exists only after the workpiece has successfully passed a quality check which completes the manufacturing process (Borgo et al., 2014).[4] In this manner, we guarantee that the product is the desired output of the activity occurrence, namely, what the designers wished to fabricate. The quality check can be implicit in the completing phase of the process or it may consist in a dedicated procedure. A manufacturing process may fail to produce the intended product due to, e.g., a failure event which stops the process or causes the item not to pass the quality test. These events can be formally modeled but they fall out of the scope of this paper. Relations $\preceq$ and $\prec$ are the usual order relations on time, whereas *endOf* is a function that, given an activity occurrence, returns the time when the occurrence ends. Finally, we also introduce a related definition (Def3) in which the product *fabricated* in the activity occurrence is not explicitly identified.

Def2 $MfgActOcc(o, x) \triangleq ActOcc(o) \wedge Product(x) \wedge PC(x, o, endOf(o)) \wedge \forall t'(t' \prec endOf(o) \rightarrow \neg PRE(x, t'))$
(o is a manufacturing activity occurrence for x means that o is an activity occurrence o, and x is a product that participates in o only at the end of o so that for all time t' preceding o's end, x is not present at time t')

Def3 $MfgActOcc(o) \triangleq \exists x MfgActOcc(o, x)$
(o is a manufacturing activity occurrence means that it is a manufacturing activity occurrence for some product x)

Figure 1 shows a taxonomic representation of the introduced classes.

2.1. Manufacturing resources and activity occurrences

Manufacturing resources are often conceived as entities that participate in manufacturing processes (Ammar-Khodja et al., 2008; Bonfatti et al., 1998; Sarkar and Šormaz, 2019). This can be formally captured as in (Def4).

[4] Since a quality test modifies an engineering quality of an entity, say changing it from unfinished to finished, one could use this fact to model an arbitrary occurrence, like a device tuning occurrence, as a manufacturing occurrence by taking the quality test to check the tuning state of the device and assuming that this transforms it into a 'new' device.

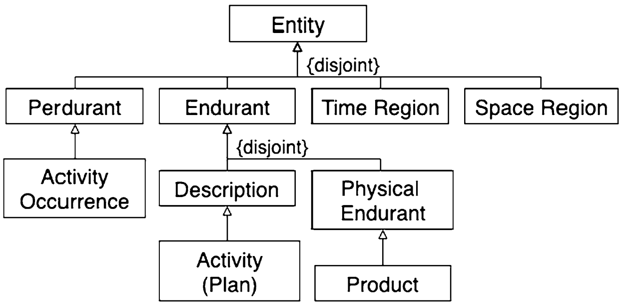

Fig. 1. Basic high-level taxonomy.

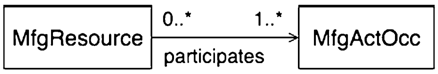

Fig. 2. First approach about manufacturing resources.

Def4 $MfgResource(r) \triangleq \exists o, t\,(MfgActOcc(o) \wedge PC(r, o, t))$
(r is a manufacturing resource means that r is an entity participating in a manufacturing activity occurrence o for some time t)

Figure 2 shows an approximation of the notion of manufacturing resource given in (Def4).

Definition (Def4) states that an entity is a resource independently of when it participates in a process, as long as such a process exists, has existed or will exist. Moreover, the product which is the outcome of a manufacturing process might be itself a resource for that very process.

The generality of this approach is sometimes restricted to capture only certain processes' participants. For instance, manufacturing resources are sometimes understood as persons or things that *add value* to a product in its production or delivery, see, e.g., the work of Qiao et al. (2011). Although it remains unclear what *adding value* means, it suggests that resources contribute to the completion of the given manufacturing process.

A further variation is adopted in works based on PSL (Grüninger, 2009b), according to which manufacturing resources are *required* for certain activities to occur; see also what proposed by Ahmad et al. (2018) and Lemaignan et al. (2006) for similar approaches. The idea is that activities cannot occur without the specified resources. In the ontology PSL Resources this is captured by axiom (Ax7), which is stated here in our notation.

Ax7 $requires(a, r) \wedge occOf(o, a) \rightarrow \exists t\,(PC(r, o, t))$
(if activity a requires r and o is the occurrence of a, then r participates in o for some time t)

The same individual resource is meant to participate in *all* occurrences satisfying the same activity, but this has undesired consequences. For instance, the theory cannot deal with cases in which the preselected resource is not available, e.g., because it is under maintenance.

To conclude, this first approach results rather limited because it heavily relies on the relation between specific resources and processes. Since in the manufacturing domain experts often need to talk of resources in planning or scheduling scenarios where there are no individual manufacturing processes to which resources can be bound, it is clear that this view needs to be extended.

2.2. Manufacturing resources and activities

In this perspective resources are conceptualized in tight connection to manufacturing activities, i.e., specifications about the activity occurrences to be realized in manufacturing contexts as discussed by, e.g., Dassisti et al. (2008), Sanfilippo et al. (2018), and Terkaj and Urgo (2012), among others. This view is more general with respect to the aforementioned approach: it weakens the direct link between resources and activity occurrences, thus enabling to model resources independently of the individual occurrences where they are meant to participate.

Definition (Def5) introduces the notion of manufacturing activity while making use of predicate *ArtifactDescr*, which specializes the predicate *Descr* for the descriptions of intentionally made entities, among which the desired outputs of activity occurrences. Looking at the formula, *PP* stands for the relationship of *proper part* in extensional mereology (Casati et al., 1999); for the sake of simplicity, *PP* is not temporalized.

Def5 $MfgActivity(a) \triangleq Activity(a) \wedge \exists x(ArtifactDescr(x) \wedge PP(x, a) \wedge \forall o \exists y(occOf(o, a) \rightarrow MfgActOcc(o, y) \wedge sat(y, x, endOf(o))))$
(a is a manufacturing activity means that a is an activity that has an artifact description x as part, and for each manufacturing activity occurrence o of a there exists a product y satisfying x at the end of o)

Axiom (Ax8) introduces the primitive relation *hasReq* between descriptions; it is informally read as x *has requirement* y, the latter standing for descriptions specifying the necessary conditions for activities to be executed. Examples include the adoption of resources with specific capabilities (e.g., drilling, loading, etc) or the fact that an activity has to be executed in compliance to certain environmental or economic policies. The relation is used in (Def6), which defines the predicate *resourceFor* between a physical endurant, a manufacturing activity, and a time span. By (Def6) and (Def7), a manufacturing resource is an endurant that satisfies (*sat*) a requirement description included in a manufacturing activity;[5] see also Fig. 3.

Ax8 $hasReq(x, y) \rightarrow Activity(x) \wedge Descr(y) \wedge PP(y, x)$
(if x has requirement y, then x is an activity and y a description that is part of x)

Def6 $resourceFor(r, a, t) \triangleq PED(r) \wedge MfgActivity(a) \wedge \exists x(hasReq(a, x) \wedge sat(r, x, t))$
(r is a resource for a at t means that r is a physical endurant, a a manufacturing activity with requirement x, and r satisfies x at time t)

Def7 $MfgResource(r)^{+} \triangleq \exists a, t(resourceFor(r, a, t))$
(r is a manufacturing resource means that r is a resource for a manufacturing activity a at some time t)

To conclude, this approach allows us to deal with the representation of manufacturing resources without committing to any specific manufacturing activity occurrence. The approach is better suited for

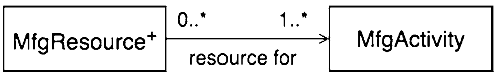

Fig. 3. Second approach about manufacturing resources.

[5](Def7) uses the superscript + in the predicate $MfgResource(r)^{+}$ to distinguish the corresponding notion of resource from the similar notions in the other approaches. Superscript are used for similar purposes across the entire paper.

planning in comparison to what discussed in Section 2.1, since it is possible to talk about manufacturing resources without binding them directly to individual occurrences.

2.3. Manufacturing resources and goals

We introduce now a further approach to represent manufacturing resources based on goal modeling, see, e.g., the work of Sanfilippo et al. (2018). The idea is that manufacturing resources are entities employed in manufacturing scenarios to bring about agents' goals, e.g., the goal of making a product with the desired characteristics. The advantage of this view over the previous ones is the possibility of modeling resources independently from the activities to which they are possibly related. For instance, one may conceive a driller as a manufacturing resource only because the driller is functional to achieve certain goals, rather than because there is an activity for a drilling process covering the description of the driller. From this point of view, the approach presented in this section weakens the one discussed in Section 2.2 to allow for resources modeling even in the absence of manufacturing activities.

Following the Belief-Desire-Intention (BDI) approach (Castelfranchi, 1998), one can understand goals as agents' mental qualities referring to *desired states of the world*. In this view, however, it can be difficult to make sense of *goal sharing* (common in manufacturing contexts), e.g., when one wishes to model multiple agents having the same goal towards the production of a certain product. An alternative way to represent goals without relying on individuals' mental states consists in objectifying and treating them as *descriptions* of desired world states (an approach proposed, e.g., by Guarino and Sanfilippo (2019) for the specific case of manufacturing knowledge representation). For instance, the goal of agent a to *have a product with characteristics b and c* is specified in a description, which is satisfied by a world state where a product with characteristics b and c exists.

Similarly to the approach in Section 2.2 to represent descriptions, axiom (Ax9) gives a minimal constraint on the relation *goalForAgent*. Looking at the axiom, *State* is a primitive predicate standing for perdurants whose (temporal) parts are all of the same type (Masolo et al., 2003). The *goalForAgent* relation is used in (Def8) to define manufacturing goal descriptions.[6]

Ax9 $goalForAgent(x, y) \rightarrow Descr(x) \wedge Agent(y) \wedge \forall s, t\,(sat(s, x, t) \rightarrow (State(s) \wedge desires(y, s)))$
(if x is a goal for agent y, then x is a description, y is an agent, and all s satisfying x at time t are such that they are states desired by y)

Def8 $MfgGoalDescr(x) \triangleq \exists y(goalForAgent(x, y) \wedge \forall s, t\,(sat(s, x, t) \rightarrow \exists v(Product(v) \wedge PC(v, s, t))))$
(x is a manufacturing goal means that x is a goal for some agent y and for each state s realizing x at t there is a product v participating in s at time t)

By looking at (Ax9) and (Def8), a manufacturing goal is satisfied by a state of the world in which there is a product that satisfies the agent's goal. Also, the formulas bind goals neither to manufacturing activities nor to their occurrences, i.e., they do not specify *how* the state and the product are obtained. (Note that one can still impose temporal constraints like, e.g., that the product does not exist before the occurrence of state s.) As said, this third approach captures the notion of resource by directly linking it to goals rather than activities or activity occurrences. Assume now that we can tie manufacturing resources to manufacturing goals (e.g., comparing the actual state and the goal via functional reasoning) and let

[6]There are two important simplifications in this part: (1) relations like *goalForAgent* and *desires* should also have a temporal parameter as an agent may change desires over time; (2) we loosely talk of agents but obviously have in mind people while playing a role (i.e., planner, designer) in manufacturing scenarios.

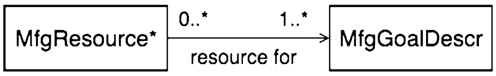

Fig. 4. Third approach about manufacturing resources.

us call such a relation *resourceFor**, then we can characterize manufacturing resources as in (Def9); see Fig. 4.

Def9 $MfgResource^*(r) \triangleq PED(r) \wedge \exists x(MfgGoalDescr(x) \wedge resourceFor^*(r, x))$
(r is a manufacturing resource means that r is a physical endurant which is a resource for a manufacturing goal x)

In order to model the link between goals and the manufacturing activity occurrences bringing them about, manufacturing activity can be defined as in (Def10). Accordingly, a manufacturing activity has a goal such that each occurrence satisfying the activity has outcome a world state satisfying the goal and therefore the product participating in the state.

Def10 $MfgActivity^*(a) \triangleq Activity(a) \wedge \exists y(MfgGoalDescr(y) \wedge PP(y, a) \wedge \forall o \exists s(occOf(o, a) \rightarrow MfgActOcc(o) \wedge sat(s, y, endOf(o))))$
(a is a manufacturing activity means that a is an activity that includes a manufacturing goal y as part, and for each occurrence o of a, o is a manufacturing occurrence and there is a state s at the end of o satisfying y)

2.4. Observations on the modeling of manufacturing resources

The three approaches presented in the previous sections are distinct views on manufacturing resources. As we have seen, the first approach (Section 2.1) is suited to model resources in tight connection to manufacturing activity occurrences. However, it is less suitable in application cases where resources are to be managed independently from manufacturing processes, as they only *possibly* participate in them. The second view (Section 2.2) deals with this issue by binding resources primarily to activities and only indirectly to activity occurrences. Accordingly, a manufacturing resource is an entity that is functional for a manufacturing activity, which may not be realized. Finally, the third approach (Section 2.3) is independent from the existence of both activities and activity occurrences, because it presents resources in direct connection to agents' goals.

All three approaches have some value. The choice of adopting one approach or the other may be pragmatic for the user depending on requirements and modeling needs. More generally, it can be argued that each approach fits a subset of factory lifecycle phases (Azevedo and Almeida, 2011): *manufacturing execution and control* (Kádár et al., 2010; Urgo and Terkaj, 2020) when the system configuration and manufacturing plan are given (Section 2.1); *detailed system design* or *production planning* (Yang et al., 2015) when the set of products is known and the process planning has been carried out (Section 2.2); *early/conceptual system design* or *strategic capacity planning* (Terkaj et al., 2019) when limited details about product and activities are available (Section 2.3).

Instead of forcing domain experts to restrict to one single approach while excluding the others, we believe that these three views should be formally and conceptually investigated and developed into a single framework. By consistently unifying the different perspectives, the intended meanings of the used notions are harmonized and, as a result, one can facilitate multiple experts to interact, e.g., via data sharing and/or data integration procedures. On the other hand, by keeping three separated views, there is no guarantee that data coming from different parties will share common meanings even if the same terms are used; notoriously, this leads to much more complex data management workflows (Grüninger, 2009a). Considering that planning and scheduling play a fundamental role in manufacturing modeling, the second approach is definitely central to this investigation and can serve as interface in the integration of the other two approaches. Thus, in the rest of the paper we focus our attention primarily on this approach with the aim of providing a unifying ontology for the representation of manufacturing resources. By integrating the three approaches, the overall conceptual and formal framework becomes more complex. However, we believe that this complexity is inevitable for manufacturing knowledge representation and data modeling where one needs to represent resources along with activity occurrences, their goals, and the underlying activities.

Finally, it must be noted that – independently from the approach one relies on – manufacturing resources are conceptualized as *roles* in all the views we have considered. Accordingly, an entity is not a manufacturing resource *per se* but it *counts as* (i.e., it *has the role of*) manufacturing resource only when certain conditions are met. For instance, when it participates in a process or it is bound to either an activity or goal. We do not however explicitly investigate the application of role theory in the paper; the reader can refer to (Loebe, 2007; Masolo et al., 2004; Mizoguchi et al., 2015) for further readings on roles modeling.

3. Manufacturing resources and activities: Deepening the model

We extend in this section the activity-centered approach of Section 2.2 to develop a theory on manufacturing resources that can serve as basis for computational implementations. Before describing the theory, Section 3.1 discusses the notions of *capability* and *capacity*, and Section 3.2 looks at activities as transformations. The formalization of the proposed approach is discussed in Section 3.3.

3.1. Capabilities and capacities

Manufacturing resources are often characterized, either explicitly or implicitly, in terms of *capabilities* and *capacities*. For instance, according to the standard ISO 15531, "a resource capability defines a group of characteristics specifying manufacturing resources under functional aspects" (ISO, 2004b, p.12). The standard relies on capabilities to constrain the participation of an entity, e.g., a device, to manufacturing activity occurrences. According to this standard, a capability is the "quality of being able to perform a given activity" (ISO, 2004a, p.5). An example is the capability of a cutting tool *to remove (a certain type of) material within certain dimensional tolerances*. Hence, the tool can be used in a cutting activity occurrence aimed at realizing a feature (e.g., hole, slot) only because of its capability. Similarly, a machining center can be characterized in terms of capability to provide a maximum spindle speed (e.g., 15,000 rpm), to control a number of axes (e.g., 3, 4 or 5 axes), to load pallets with specific dimensions (e.g., tables 400 × 400 mm, 500 × 500 mm), etc.

In the same standard, capacity is defined as the "measure of the quantity of product (or component) a resource can process per unit of time" (ISO, 2004b, p.24). Although it is hard to find explicit definitions of notions like capability and capacity in the literature about manufacturing, the ISO 15531 view seems to be shared across the domain (Järvenpää et al., 2019; Sarkar and Šormaz, 2019; Solano et al., 2016).

Focusing on capabilities, from the comments above it is clear that ISO 15531 understands them as resources' functionalities, i.e., what resources – in virtue of the way in which they have been designed – are able to do. The notion of functionality is differently understood in engineering (Erden et al., 2008) and, as a consequence, differently represented in ontologies (Garbacz et al., 2011). It is beyond the scope of this paper to present an overview on these alternative approaches on functionality (see the work of Borgo et al. (2011) and references in it). They can differ quite considerably going from a behavioral perspective to a relational characterization or transformational view. Unifying approaches of engineering function theories exist as well (Mizoguchi et al., 2016; Kitamura and Mizoguchi, 2013), and these can be exploited for integrating the resource-based view proposed in this paper.

In recent works about manufacturing modeling based on the Basic Formal Ontology (Arp et al., 2015), Otte et al. (2019) and Sarkar and Šormaz (2019) propose to conceive and represent capabilities in terms of BFO's *dispositions*. Leaving aside the philosophical debate on what these latter entities are (Mumford, 2003), in applied ontology they are commonly understood as objects' characteristics which are *manifested* only if specific circumstances occur (Azevedo et al., 2015). For instance, the capability of a magnet to attract metal characterizes the magnet whenever the latter exists. However, as a disposition, it is manifested only when the magnet is close enough to a piece of metal and, therefore, it may never manifest. Alternatively, Borgo and Vieu (2009) characterize artifacts' capabilities by relying on the notion of (individual) quality provided in DOLCE (Borgo and Masolo, 2009). The difference is not only theoretical but practical as well, since there are different modeling consequences. Approaches based on dispositions, like (Azevedo et al., 2015; Otte et al., 2019; Sarkar and Šormaz, 2019), require the introduction of both the manifestation trigger and the surrounding conditions: the knowledge engineer is driven to develop a system which is conceptually rich and formally complex. On the other hand, the approach based on individual qualities, like (Borgo and Vieu, 2009; Borgo et al., 2014), allows for a light modeling: one can introduce the capabilities without further conditions leaving to the knowledge engineer the choice of which information to include based on the relevant scenarios and needs.

Turning to capacities and following ISO 15531, note the similarity with capabilities: the latter model *what* a resource is able to do (e.g., to cut material at a certain speed), the former *how many* things it does (e.g., a polished threaded hole each 5 minutes). Therefore, capacities can be modeled as resources' *relational* qualities making explicit the parameters' ranges that a resource is able to satisfy in a given situation.

3.2. Activities as descriptions of transformations

The formalization of the approach to resources based on activities requires a shared understanding of what we mean by activity. In this paper a (manufacturing) activity a consists of a sequence of pairs $\langle s_i, a_i\rangle$ where s_i is a state description and a_i a primitive or atomic activity.[7] In particular, this means that our activities are linear, in the sense that neither branches nor arbitrary loops are allowed. This choice is clearly an oversimplification that we make to keep the presentation formally simple. Another simplifying assumption is that all the activities that we mention are realizable: given a state satisfying s_i the execution of occurrence o_i, such that $occOf(o_i, a_i)$, is possible and, moreover, at the time the occurrent o_i is complete the world is in a state satisfying s_{i+1}. Clearly, the initial state s_0 for an activity a must hold at the time the activity a starts to be executed, i.e., when an occurrence o_0 that makes $occOf(o_0, a_0)$ true, is intended as part of the execution of activity description a. The reader might find helpful to read occurrence o_i as a process that satisfies the description of a manufacturing transformation.

[7]We assume that a list of primitive activities is given and that it suffices to generate all activities one needs to use in the modeling scenario (or for the purpose) at stake. Note that we do not commit to further requirements about granularity or coverage of the given activities. In addition, recall that the choice of conceiving activities as (descriptions of) transformations is compatible with PSL, see, e.g., the PSL theory *disc* (discrete) *state* (Grüninger, 2009b).

Let activity a be the finite sequence $\langle s_0, a_0\rangle \dots \langle s_n, a_n\rangle$ and assume that the world is in a state satisfying the state description s_0 (the initial state relatively to that activity). An execution of the activity is a sequence of exactly n activity occurrences $o_0, o_1, \dots o_n$ such that for each i, formula $occOf(o_i, a_i)$ holds, and at the completion of o_i, for $i < n$, state description s_{i+1} holds.

Generally speaking, one can follow the modeling approach developed in this section even adopting a broader notion of activity, for instance to generalize activities beyond PSL. This can be achieved by lifting some conditions like the following:

(1) an activity is not necessarily linear (e.g., it can be a directed acyclic graph or even include loops);
(2) an activity is not necessarily composed of primitive activities;
(3) completion of occurrence o_i is not guaranteed;
(4) completion of occurrence o_i might not satisfy s_{i+1};
(5) activities do not need to be finite (e.g. continuous casting).

However, these generalizations require further formal constraints to ensure that the relationships between activities, activity occurrences, states, and state descriptions are as expected. The representation of such constraints goes beyond the scope of this paper.

3.3. A unified approach

Recall that it is useful in manufacturing to distinguish process plans from production plans. A *process plan* defines the set of work instructions to convert a part from its initial form to the required final form (Kamrani and Vijayan, 2008). The process plan may include the description of manufacturing processes, operational setup, process parameters, and possibly equipment and/or machine tool selection. On the other hand, based on product demand/orders, a *production plan* 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 (Chryssolouris, 1992). Even though our attention focuses on process planning and the relation between resources and process plans, these definitions are fundamental also for production planning. Indeed, a production plan incorporates one or more process plans and resolves possible ambiguities in the process plan. Both process and production planning can be hierarchically addressed.

In order to integrate in a unified approach the three views discussed in Section 2 along with the representation of both capabilities and manufacturing activities, the latter understood as transformations, we first extend the predicate *Descr*. Axiom (Ax10) lists three classes of descriptions. Axiom (Ax11) specializes artifact descriptions (see Section 2.2) to either amounts of matter (*AoM*) or physical objects (*POB*; see below for comments on the latter two notions). This because when we talk of individual artifacts, one may want to refer to either physical objects or amounts of matter that are intentionally produced; an example of the latter is an amount of concrete or an amount of some minerals resulting from mining. *MgfCapabilityDescr* captures the descriptions of capabilities, which are satisfied by qualities (*Qlt*) inhering in (I) physical endurants, see (Ax12). *StateDescr* models the description of world states (Ax13). The notions of quality, physical object, and amount of matter are well-known in applied ontology and they are similarly conceived across different foundational ontologies like DOLCE (Masolo et al., 2003) or the Unified Foundational Ontology (UFO; Guizzardi et al., 2015). We do not report here their formal representation limiting ourselves to a few aspects related to our topic.

Qualities are entities qualifying and depending on their bearers, e.g., the weight-quality of a driller. They are also specific to their bearers: the weight-quality of driller d_1 is different from the weight-quality of driller d_2, no matter how similar the drillers are. Qualities can be distinguished according to resemblance criteria which help to model qualities in disjoint classes like weight-qualities, color-qualities, size-qualities, speed-qualities, and so on. Qualities can be organized in *quality spaces*, namely, structures like metric spaces where qualities' values are provided. For instance, saying that two drillers weigh 8 kg means that the weight-quality of each of them is associated with the same location (namely, the location corresponding to value 8) in the weight-quality-space when measured according to similar measurement methodologies and the same unit (namely, kilogram). Capability spaces are similar but usually have a richer structure since they may have several dimensions and include constraints due to physical laws. Concerning amounts of matter (e.g., some amounts of metal powder or liquids), recall that – differently from material objects like a drill or a plank of wood – their identity is bound to their parts, therefore they do not survive parts' loss or change (Masolo et al., 2003).

Ax10 $ArtifactDescr(x) \vee MfgCapabilityDescr(x) \vee StateDescr(x) \rightarrow Descr(x)$
(if x is an artifact description or a manufacturing capability description or a state description, then x is a description)

Ax11 $ArtifactDescr(x) \wedge sat(y, x, t) \rightarrow AoM(y) \vee POB(y)$
(if x is an artifact description and y satisfies x at time t, then y is either an amount of matter or a physical object)

Ax12 $MfgCapabilityDescr(x) \wedge sat(y, x, t) \rightarrow Qlt(y) \wedge \forall z(I(y, z) \rightarrow PED(z))$
(if x is a capability description and y satisfies x at time t, then y is a quality such that for all z in which y inheres, z are physical endurants)

Ax13 $StateDescr(x) \wedge sat(y, x, t) \rightarrow State(y)$
(if x is a state description and y satisfies x at time t, then y is a state)

Definitions (Def11)–(Def14) are introduced to re-define the notion of manufacturing activity in a way that better unifies the approaches in Section 2 and Section 3.2. (Def11) defines the relation *hasGoal* between an activity and a state description.[8] The axiom establishes a correspondence with the predicate *goalForAgent* introduced by (Ax9) in Section 2.3. It therefore assumes the existence of an agent who attributes a goal to an activity. (Def12) introduces the relation *hasOutput* to explicitly capture the intended output of activities; $hasOutput(x, y, z)$ is read as *activity x has intended output y in state z*, where the variable z stands for the description of the activity's goal. As before, one could add that this condition is never true before the related activity occurrence is complete. Both (Def13) and (Def14) extend the relation *has requirement* (*hasReq*) introduced in Section 2.2. The former definition models the relation *has resource requirement* (*hasResourceReq*) to capture the requirements about resources included in activities. The latter is used to capture the relation *has initial state requirement* (*hasInitialStateReq*). Following Section 3.2, we make sense in this way of the fact that each manufacturing activity comprises a state (description) making explicit the state of the world for its occurrence to take place.

Def11 $hasGoal(x, y) \triangleq Activity(x) \wedge StateDescr(y) \wedge PP(y, x) \wedge \exists z(goalForAgent(y, z)) \wedge \forall o \exists s(occOf(o, x) \rightarrow sat(s, y, endOf(o)))$
(x has goal y means that 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)

Def12 $hasOutput(x, y, z) \triangleq hasGoal(x, z) \wedge ArtifactDescr(y) \wedge PP(y, x) \wedge \forall o\, \exists s, p(occOf(o, x) \rightarrow sat(p, y, endOf(o)) \wedge sat(s, z, endOf(o)) \wedge PC(p, s, endOf(o)))$

[8]Similarly to the Section 2.3, we treat goals as descriptions of desired states.

(x has intended output y in z means that x has goal z and y is an artifact description 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)

Def13 $hasResourceReq(x, y) \triangleq hasReq(x, y) \wedge (MfgCapabilityDescr(y) \vee ArtifactDescr(y))$

(x has resource requirement y means that x has requirement y and y is either a capability description or an artifact description)

Def14 $hasInitialStateReq(x, y) \triangleq hasReq(x, y) \wedge StateDescr(y) \wedge \forall o \exists s(occOf(o, x) \rightarrow sat(s, y, beginOf(o)))$

(x has initial state requirement y means that x has requirement y, which is a state description, and for all occurrences o of x, there is a state s satisfying y at the beginning of o)

These relations are used in (Def15) to define manufacturing activity.

Def15 $MfgActivity^{\dagger}(a) \triangleq \exists y, z, w, v(hasInitialStateReq(a, y) \wedge hasResourceReq(a, v) \wedge hasOutput(a, z, w))$

(a is a manufacturing activity means that a has some initial state requirement y, some resource requirement v, and some output z in state w)

For the sake of clarity, let us comment on (Def15). First, following the discussion in Section 3.2, the requirement on the initial state description establishes the conditions of the world state in order for the activity to be executed. A state description could be further characterized to describe the world state to be present for the beginning of an activity occurrence (e.g., a manufacturing environment at controlled temperature). Second, to match the goal-view of Section 2.3, the output of an activity is now associated to a goal description. Third, the resource requirement is needed to characterize resources in tandem with manufacturing activities. Accordingly, (Def16) slightly revises the definition (Def6) given in Section 2.2, where *hasResourceReq* is now used to explicitly qualify the descriptions that a physical endurant has to satisfy to be considered as manufacturing resource.

Def16 $resourceFor^{\dagger}(r, a, t) \triangleq PED(r) \wedge MfgActivity^{\dagger}(a) \wedge \exists x(hasResourceReq(a, x) \wedge sat(r, x, t))$

(r is a resource for a at t means that r is a physical endurant, a a manufacturing activity with resource requirement x and r satisfies x at time t)

This new predicate is used in (Def17) to define manufacturing resource by revising (Def7).

Def17 $MfgResource^{\dagger}(r) \triangleq \exists a, t(resourceFor^{\dagger}(r, a, t))$

(r is a manufacturing resource means that r is a resource for a at some time t)

In the spirit of Section 2.1, we need to bind resources to manufacturing activity occurrences but this time in tandem with manufacturing activities and their requirements. Accordingly, axiom (Ax14) establishes that for a resource to participate in a manufacturing activity occurrence, it must satisfy the resource requirements specified in the activity that the occurrence realizes.

Ax14 $MfgResource^{\dagger}(r) \wedge MfgActOcc(o) \wedge PC(r, o, t) \rightarrow \exists a, x(MfgActivity^{\dagger}(a) \wedge occOf(o, a) \wedge hasResourceReq(a, x) \wedge sat(r, x, t))$

(if a manufacturing resource r participates in manufacturing activity occurrence o at time t, then r satisfies the resource requirement x of some manufacturing activity a that o realizes)

4. Case study: Design of manufacturing systems

We present in this section an industrial use case related to the design of manufacturing systems where the modeling of manufacturing resources plays a key role. The use case deals with the design of an assembly line, namely, a powertrain valve assembly on a cylinder head, as described by Ascheri et al. (2014), Ascheri (2016), and Belkadi et al. (2018). An ontology-based approach supports the integrated factory design by providing a shared common model that can consistently evolve during the design phase. The following input data are given in the case study at the design phase: (*i*) The process plan of the product to be assembled; (*ii*) the set of sequential activities decomposing the process plan; (*iii*) the input component or material for each activity; (*iv*) the requirements to execute each activity.

The system design phases involve several decisions, e.g., the selection of production resources to execute activities (i.e., line configuration), the grouping of activities into production stages (i.e., line balancing), as well as the optimization of the capacity of inter-operational buffers (i.e., buffer allocation problem). We assume that a proper capability assessment has already been completed and each activity can be executed by an automatic solution consisting of resources matching the requirements; alternatively, a manual solution is possible in some cases. Note that automatic and manual solutions require different sets of manufacturing resources. For example, manual assembly is carried out by human operators usually with the aid of simple power tools, whereas automatic assembly requires specific pieces of equipment for robotic handling and processing (e.g., fixtures and end effectors).

Table 1 reports some relevant data defining the use case, namely, (*i*) the list of activities (ID and description); (*ii*) input components for each activity (WIP stands for 'work in progress', i.e., the result from the previous activity); (*iii*) resources needed to execute each activity. Each resource can be further characterized in terms of failure modes, investment and operating cost, size, etc. The processing time may depend on the resources employed to execute the activities.

Even though the process plan is linear, various feasible system configurations can be designed based on the requirements and the selectable resources. Line balancing and line configuration will be carried out by considering objective function and constraints related to investment cost, operating cost, throughput, among other criteria (Belkadi et al., 2018). After line balancing has defined the assignment of activities to production stages (i.e. workstations), each workstation must include the super-set of production resources needed for all activities to be performed in that stage.

We discuss in the next section how the case study can be represented by means of both an existing approach and the ontology presented in Section 3.3 while stressing the differences between the two approaches.

5. Modeling the case study

We show in Section 5.1 the representation of the case study presented in the previous section by means of an existing ontology for manufacturing data modeling (Terkaj et al., 2019). By stressing the limits of this approach, we will present in Section 5.2 an alternative representation based on the theory discussed in Section 3.3. For the sake of the example, we consider only the first activity in Table 1, i.e., op_{10}; this should suffice to understand how to model the rest of the table as well.

5.1. An IFC-based representation

The ontology used in this subsection provides an extensible representation of factory entities related to production systems, resources, processes, and products (Terkaj et al., 2019). It is formally represented

Table 1
Use case data for powertrain valve assembly on a cylinder head

Activity ID	Activity description	Input components	Activity requirements	Resource in Automatic solution	Resource in Manual solution
op10	Load cylinder head on pallet	Raw cylinder head	loading workholding	palletizing robot pallet	operator, hoist pallet
op20	Identify cylinder head	WIP cylinder head	handling tooling workholding	robot vision system pallet	operator hand-held vision system pallet
op30	Apply sealant and lubricant	WIP cylinder head	handling tooling workholding	robot sealant dispensing tool pallet	operator hand-held sealant dispensing tool pallet
op40	Install intake and exhaust valves	WIP cylinder head, intake and exhaust valves	handling tooling workholding	robot handling gripper pallet	operator power tool pallet
op50	Valve blow-by leak test	WIP cylinder head	handling tooling workholding	robot leak test tool pallet	N/A N/A N/A
op60	Rollover	WIP cylinder head	handling workholding	rollover equipment pallet	N/A N/A
op70	Assemble valve stem seal	WIP cylinder head, valve stem seals	handling tooling workholding	robot handling gripper pallet	operator power tool pallet
op80	Press valve stem seals	WIP cylinder head	handling tooling workholding	robot pressing tool pallet	operator power tool pallet
op90	Assemble valve springs	WIP cylinder head, valve springs	handling tooling workholding	robot handling gripper pallet	operator power tool pallet
op100	Assemble valve spring retainer	WIP cylinder head, valve spring retainer	handling tooling workholding	robot handling gripper pallet	operator power tool pallet
op110	In process verification	WIP cylinder head	handling tooling workholding	robot verification tool pallet	N/A N/A N/A
op120	Unload cylinder head assembly	WIP cylinder head	unloading workholding	palletizing robot pallet	operator, hoist pallet

as a modular OWL ontology based on international standards such as ISO 16739 (IFC; Liebich et al., 2013) and the W3C SSN/SOSA (Janowicz et al., 2019). For the purposes of the paper, we focus on the modules named *IFC4_ADD1*, *IFC4_ADD1_extension*, and *factory* (see Table 2). The first module is automatically generated from IFC (Pauwels et al., 2017), and provides a high-level characterization for classes such as product, process, and resources. The second module integrates and extends IFC4_ADD1.

Table 2

Prefixes of ontology modules that are available online

Prefix	Prefix IRI of ontology module
ifc	http://ifcowl.openbimstandards.org/IFC4_ADD1#
ifcext	http://www.ontoeng.com/IFC4_ADD1_extension#
fa	http://www.ontoeng.com/factory#

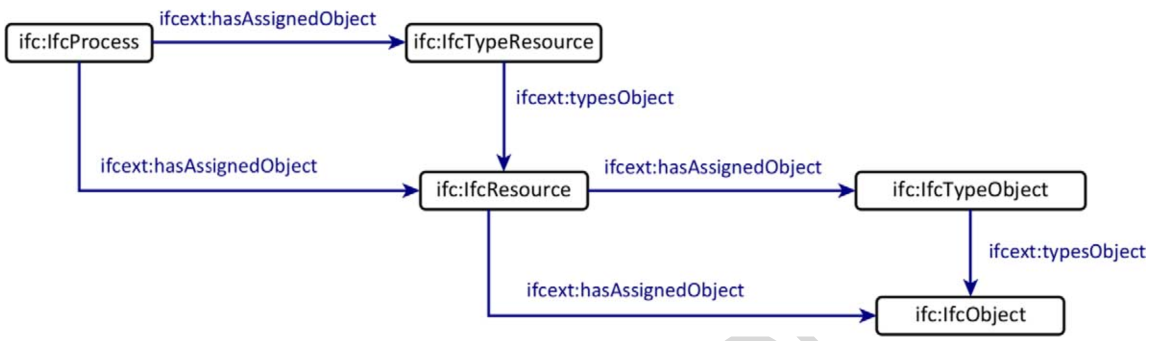

Fig. 5. Modeling pattern in the OWL factory data model partially based on the IFC standard.

In particular, this extension simplifies the representation of relations between classes. Recall indeed that IFC heavily relies on the use of reified relations (Borgo et al., 2015), an approach often resulting in verbose and computationally expensive (e.g., in terms of memory) data models. We therefore adopt here a simplified version of IFC, similarly to what proposed by de Farias et al. (2015). Finally, the third module further extends IFC4_ADD1 to the manufacturing domain.

Figure 5 shows an excerpt of the ontology focused on the relation between processes and resources. A process (*ifc:IfcProcess*) is an event that transforms input products into output products while making use of resources under specific control rules. A resource (*ifc:IfcResource*) represents an entity that is needed to perform the process. In turn, various objects, such as physical products, people, and materials, can be employed as resources.

The class *ifc:IfcTypeObject* (and similarly the class *ifc:IfcTypeResource* since in IFC the latter is a subclass of the first) can be used to define "specific information about a type, being common to all occurrences of this type" (Liebich et al., 2013). Instances of *ifc:IfcTypeObject* are represented by sets of properties which apply to all the associated instances of *Ifc:IfcObject* and its subclasses. Object types can be however directly instantiated without being assigned to instances of *Ifc:IfcObject* (Liebich et al., 2013; Borgo et al., 2015). This approach is useful whenever the user needs to be generic or non-committal while defining the resources needed or used to execute a process.

The object property *ifcext:hasAssignedObject* is employed to represent both the relation between a process and the needed resources, and the relation between a resource and the objects that are used as resources. The object property *ifcext:typesObject* represents the relation between object types and object occurrences. Note that the properties *ifcext:hasAssignedObject* and *ifcext:typesObject* replace the use of the reified relationships *ifc:IfcRelAssigns* and *ifc:IfcRelDefinesByType*, respectively, defined in IFC.

By using this ontology, the use case presented in Section 4 can be instantiated as shown in Fig. 6 for activity op_{10}. The model makes use of additional classes, namely, *fa:ManufacturingTask*, *fa:ProductionResource*, and *fa:ProductionResourceType*, which are subclasses of *ifc:IfcProcess*, *ifc:Ifc-*

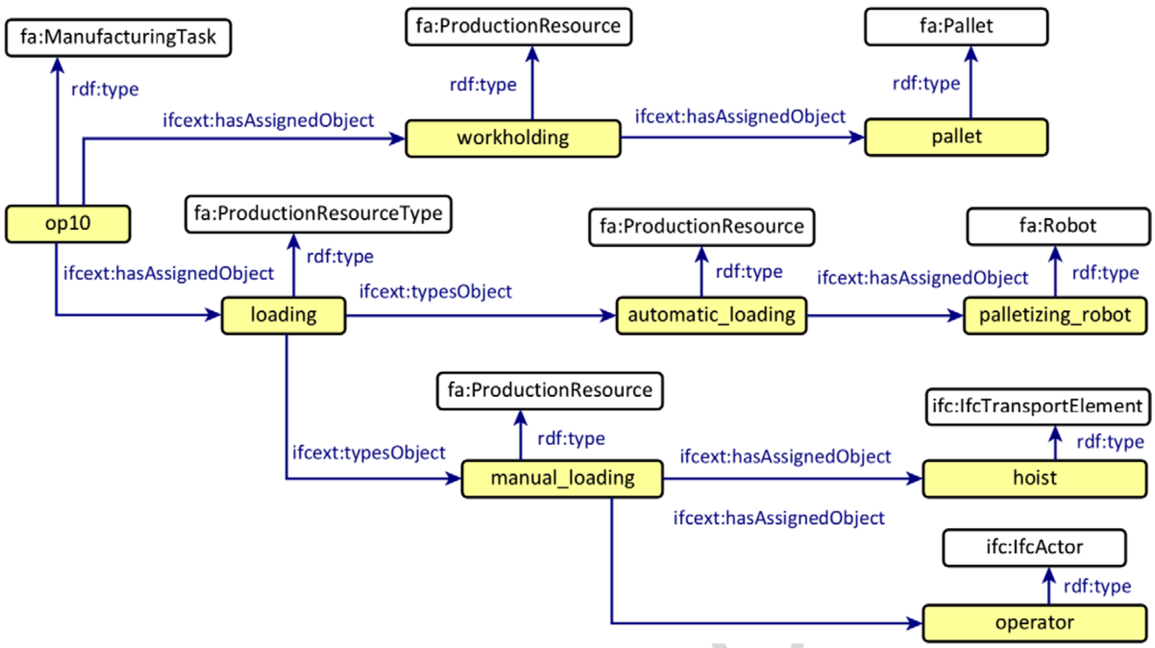

Fig. 6. Instantiation of activity op_{10}.

Resource, and *ifc:IfcTypeResource*, respectively. Furthermore, *fa:Robot*, *fa:Tool*, *fa:Pallet*, *ifc:IfcActor*, *ifc:IfcTransportElement* are all subclasses of *ifc:IfcObject*.

By looking at Table 1, activity op_{10} needs a workholding resource, which is provided by a pallet both in the automatic and manual solution. In addition, a *loading* resource is needed to place the cylinder head on the pallet. The loading resource belongs to the class *fa:ProductionResourceType* so that it is possible to link it with two options of resources, i.e. the automatic and manual loading resource, via the type/occurrence relation. The automatic loading resource uses a palletizing robot, whereas the manual loading resource is provided by combining an operator and a hoist. This example demonstrates how this modeling pattern enables the definition of alternative resources that can be later resolved, either during the design or planning phase.

Finally, after the instantiation of the use case, it is possible to analyze the resulting OWL ontology with respect to the approach presented in Section 3:

- The ontology presented in this subsection heavily relies on the use of IFC and inherits the distinction between classes like *ifc:IfcObject* (or *ifc:IfcResource*) and *ifc:IfcTypeObject* (*ifc:IfcTypeResource*). Although the distinction is useful in applications, the intended meanings of these classes are not fixed in the standard and they therefore remain open to alternative interpretations. For instance, both *ifc:IfcTypeResource* and *ifc:IfcResource* can be used to model *descriptions of* resources but at different levels of granularity; therefore, in order to develop computational ontologies in a homogeneous manner, one should decide how to use them and properly distinguish between their instances.
- The definitions of classes *ifc:IfcTypeResource* and *ifc:IfcResource* are ambiguous in the IFC standard and there is a lack of clarifying examples. Herein, these classes are used with (at least) two different meanings, namely, to model either a generic resource description or the description of capability requirements. However, descriptions of resources and descriptions of capabilities need to

be properly distinguished to enhance the transparency of both computational ontologies and data, as proposed in Section 3.

- A better approach is needed to specify resources' capabilities. Looking, e.g., at Fig. 6, the same relation *ifcext:hasAssignedObject* is used to link activities to either resource descriptions or capability descriptions, as well as to make explicit the relation between capabilities and resources (see, e.g., the link between *workholding* and *pallet*). However, these two interpretations should not be confused. In the approach presented in Section 3.3, the first relation corresponds to the predicate *has resource requirement* (*hasResourceReq*) holding between descriptions; the second one corresponds to *satisfies* (*sat*); this distinction makes clear that physical resources must bear certain capability/qualities.
- The ontology used in this section lacks a concept similar to the predicate *satisfies* (*sat*) between descriptions like resource requirements and the corresponding physical objects, among other types of entities. For instance, the relation between *automatic_loading* and *palletizing_robot* is directly and explicitly introduced by the user (e.g., system designer, process planner). An assessment of the matching would require an extension of the ontology to represent qualities of the object and capability/resource requirements.
- The presence of alternative solutions is modeled by exploiting the type/occurrence relation (see Fig. 6). However, this possibility is not explicitly considered in the reference standard and a more formal representation would be beneficial.

5.2. *Repesentation based on the unified approach*

We now adopt and extend the approach presented in Section 3.3 to model the use case. In the following, Table 1 is interpreted as follows. Column 1 (Activity ID) lists constants while column 2 (Activity description) lists descriptions of processes. The latter are represented as classes of manufacturing activities; e.g., the first entry – *Load Cylinder Head on Pallet* – refers to the activity for loading raw parts, see (Ax15).[9] Also, note that the table implicitly assumes a distinction between simple and non-simple activities; e.g., each activity listed in the table, when taken at a certain granularity, does not comprise other activities as proper (temporal) parts, differently from the whole activity comprising all operations from op_{10} to op_{120}. Column 3 (Input components) lists input requirements, i.e., components that are manipulated during a manufacturing activity occurrence to fabricate products. This is captured by definition (Def18) with the notion of *component requirement* (*hasComponentReq*). On the other hand, the output of activities $op_{10} \dots op_{110}$ consists in the *WIP cylinder head* in a state specified by the completion of the activity itself. For example, the output of op_{10} is the *WIP cylinder head* in the state *loadedOnPallet*. The output of activity op_{120} is the finished product, i.e. the *cylinder head*. Column 4 (Activity requirements) lists requirements on resource capabilities. They can be therefore represented via (Def13). Finally, columns 5 and 6 (Resource in Automatic solution, Resource in Manual solution) list the descriptions of resources either in the automatic or manual configuration that can execute the activity occurrences. As said, when an activity is associated to both an automatic solution and a manual solution, the corresponding activity occurrence can be executed by both.

Ax15 $LoadCylinderHead(x) \rightarrow MfgActivity^{\dagger}(x)$
(if x is a load cylinder head, then x is a manufacturing activity)

[9]Further formulas can be used to cover all activities in Table 1, as well as to further characterize their intended meaning.

Def18 $hasComponentReq(x, y) \triangleq hasReq(x, y) \wedge \exists z, s(hasOutput(x, z, s) \wedge \forall o \exists v, p(occOf(o, x) \rightarrow sat(p, z, endOf(o)) \wedge sat(v, y, endOf(o)) \wedge P(v, p, endOf(o))))$
(x has component requirement y means that x has requirement y and output z in s, and for all occurrences o of x, there exist a p and a v at the end of o satisfying z and y, respectively, such that v is part of p)

Formulas (f1)–(f5) represent the first operation (op_{10}) in Table 1.[10] Formulas (f4) and (f5) represent the match between the capability requirements and the descriptions of the resources whose instances could be used to realize op_{10}. Since the case study assumes that the planner has not made a choice on the resources to use, resource descriptions are not directly linked to op_{10}.

f1 $LoadCylinderHead(op_{10}) \wedge hasComponentReq(op_{10}, rawCylinderHead) \wedge hasOutput(op_{10}, rawCylinderHead, loadedOnPallet)$
(op_{10} is a load cylinder head with component requirement *rawCylinderHead* and output *rawCylinderHead* in the state of *loadedOnPallet*)
f2 $hasResourceReq(op_{10}, loading) \wedge hasResourceReq(op_{10}, workholding) \wedge MfgCapabilityDescr(loading) \wedge MfgCapabilityDescr(workholding)$
(op_{10} has *loading* and *workholding* as capability requirements)
f3 $ArtifactDescr(pallet) \wedge sat(pallet, workholding, t_{10})$
(*pallet* is an artifact description satisfying the *workholding* capability at time t_{10})
f4 $ArtifactDescr(palletizingRobot) \wedge sat(palletizingRobot, loading, t_{10})$
(*palletizingRobot* is an artifact description satisfying the *loading* capability at time t_{10})
f5 $Descr(operator) \wedge ArtifactDescr(hoist) \wedge sat(operator + hoist, loading, t_{10})$
(*operator* and *hoist* together satisfy the *loading* capability at time t_{10})

Note that differently from the IFC-based approach presented in Section 5.1, we are now able to explicitly distinguish between the entities involved in the case study and the relations holding between them. For example, the distinction between resource and capability descriptions are explicitly stated; also, relations like *has resource requirements*, *satisfies*, *has component requirement*, and *has output* are useful to provide a clear representation of the case study.

6. Conclusions

The research work presented in the paper is motivated by the need of investigating the foundational grounds for resource modeling in manufacturing. We started from three existing high-level approaches which are well motivated from both an ontological and manufacturing stance. The choice of relying on one or the other depends on the modeling requirements that an ontology should satisfy, as well as on the level of abstraction one wishes to achieve. Our contribution has been presented in Section 3, where the main features of the three approaches are unified in a single framework. In this way, end-users have the possibility of representing manufacturing resources by taking into account activities, goals, and activity occurrences happening in time. Hence, one can model application scenarios within or across various phases of factory lifecycle, e.g., design, planning, scheduling, or execution.

Further work on the treatment of resources is needed at both the foundational and application levels. First, we have tacitly assumed that resources are organized in different high-level classes, e.g., resources

[10]We label formulas relative to the example with f. Also, we use t_{10} for the time in which operation op_{10} is executed.

that execute manufacturing processes and resources that undergo processes, among others. The development of a taxonomy of resources is therefore needed to classify and characterize them in a proper manner. Second, as we have seen in Section 1, resources are sometimes understood in connection to ownership or availability conditions. Clearly, a resource has to be available in a specific environment for it to be used. Also, ownership of resources is particularly relevant in modern scenarios (e.g. Cloud Manufacturing) where one has the possibility of using third-party technologies. Therefore, characterizing the business dimension of resources can make a difference for application settings. Third, we gave only a quick look at the modeling of resources' capabilities. This topic needs to be further investigated to be suitable for implementation. A comparison with theories about engineering functionalities is required to understand the differences and similarities between functionalities and capabilities. Also, since capabilities can be more or less complex (Järvenpää et al., 2019), a robust formal approach for their representation is needed.

Finally, the theory presented in this paper gives the backbone of our approach. While we showed how it works, only a properly extended and refined version (in some computational logic) can be used to handle data in engineering applications and, possibly, to exploit automatic reasoning over data.

Acknowledgements

We wish to thank Nicola Guarino, Lorenzo Solano, Pedro Rosado, Fernando Romero, and Marcello Urgo for previous discussions on the modeling of manufacturing resources leading to this paper. We also thank the reviewers for their useful comments.

Appendix

Table 3

Overview of the predicates used in the paper

Predicate	Informal reading	Source
$Activity(x)$	x is an activity	PSL
$ActOcc(x)$	x is an activity occurrence	PSL
$Agent(x)$	x is an agent	–
$AoM(x)$	x is an amount of matter	DOLCE
$ArtifactDescr(x)$	x is an artifact description	–
$beginOf(x)$	the beginning time of perdurant x	–
$Descr(x)$	x is a description	–
$desires(x, y)$	agent x desires the state satisfying y	–
$ED(x)$	x is an endurant	DOLCE
$endOf(x)$	x is the ending time of perdurant x	–
$goalForAgent(x, y)$	x is the description of a goal desired by agent y	–
$hasComponentReq(x, y)$	x is an activity with component requirement y	–
$hasGoal(x, y)$	x is an activity with goal y	–
$hasInitialStateReq(x, y)$	x is an activity with initial state requirement y	–
$hasOutput(x, y, z)$	x is an activity with output y in state z	–
$hasReq(x, y)$	x is an activity with requirement y	–
$hasResourceReq(x, y)$	x is an activity with resource requirement y	–

Table 3
(Continued)

Predicate	Informal reading	Source
$I(x, y)$	x is a quality inhering in y	DOLCE
$in(x, r, t)$	x is located in space region r at time t	–
$LoadCylinderHead(x)$	x is an activity for loading cylinder heads	–
$occOf(x, y)$	x is an activity occurrence satisfying activity y	PSL
$MfgActivity(x)$, $MfgActivity^{\star}(x)$, $MfgActivity^{\dagger}(x)$	x is a manufacturing activity	–
$MfgActOcc(x)$	x is a manufacturing activity occurrence	–
$MfgActOcc(x, y)$	x is a manufacturing activity occurrence with output y	–
$MfgCapabilityDescr(x)$	x is the description of a manufacturing capability	–
$MfgGoalDescr(x)$	x is a manufacturing goal description	–
$MfgResource(x)$, $MfgResource^{\star}(x)$, $MfgResource^{\dagger}(x)$, $MfgResource^{+}(x)$	x is a manufacturing resource	–
$PC(x, y, t)$	x participates in y at t	DOLCE
$PD(x)$	x is a perdurant	DOLCE
$PED(x)$	x is a physical endurant	DOLCE
$POB(x)$	x is a physical object	DOLCE
$PRE(x, t)$	x is present at time t	DOLCE
$P(x, y, t)$	x is part of y at t	mereological relation
$PP(x, y)$	x is proper part of y	mereological relation
$Product(x)$	x is a product	–
$Qlt(x)$	x is an individual quality	DOLCE
$resourceFor(x, y, t)$, $resourceFor^{\dagger}(x, y, t)$, $resourceFor^{\star}(x, y)$	x is a physical endurant satisfying a description in the manufacturing activity y (at time t)	–
$requires(x, y)$	activity x requires resource r	PSL
$sat(x, y, t)$	x satisfies the description y at time t	–
$SpaceRegion(x)$	x is a space region	–
$State(x)$	x is a state	DOLCE
$StateDescr(x)$	x is the description of a state	–
$T(x)$	x is a time span	–
$\prec, \preceq$	ordering relations between time spans	–
$+$	mereological sum	mereological relation

References

Ahmad, M., Ferrer, B.R., Ahmad, B., Vera, D., Lastra, J.L.M. & Harrison, R. (2018). Knowledge-based PPR modelling for assembly automation. *CIRP Journal of Manufacturing Science and Technology*, *21*, 33–46. doi:10.1016/j.cirpj.2018.01.001.

Alting, L. (1982). *Manufacturing Engineering Processes*. Marcel Dekker.

Ammar-Khodja, S., Perry, N. & Bernard, A. (2008). Processing knowledge to support knowledge-based engineering systems specification. *Concurrent Engineering*, *16*(1), 89–101. doi:10.1177/1063293X07084642.

Arp, R., Smith, B. & Spear, A.D. (2015). *Building Ontologies with Basic Formal Ontology*. Mit Press.

Ascheri, A. (2016). Powertrain assembly lines automatic configuration using a knowledge based engineering approach. PhD thesis, Politecnico di Torino.

Ascheri, A., Colombo, G., Ippolito, M., Atzeni, E. & Furini, F. (2014). Feasibility of an assembly line layout automatic configuration based on a KBE approach. In *Proceedings of the 2014 International Conference on Innovative Design and Manufacturing (ICIDM)* (pp. 324–329). doi:10.1109/IDAM.2014.6912715.

Azevedo, A. & Almeida, A. (2011). Factory templates for digital factories framework. *Robotics and Computer-Integrated Manufacturing*, *27*(4), 755–771. Conference papers of Flexible Automation and Intelligent Manufacturing. doi:10.1016/j.rcim.2011.02.004.

Azevedo, C.L., Iacob, M.-E., Almeida, J.P.A., van Sinderen, M., Pires, L.F. & Guizzardi, G. (2015). Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for ArchiMate. *Information systems*, *54*, 235–262. doi:10.1016/j.is.2015.04.008.

Belkadi, F., Colledani, M., Urgo, M., Bernard, A., Colombo, G., Borzi, G. & Ascheri, A. (2018). Modular design of production systems tailored to regional market requirements: A frugal innovation perspective. *IFAC-PapersOnLine*, *51*(11), 96–101. doi:10.1016/j.ifacol.2018.08.241.

Berardi, D., Calvanese, D. & De Giacomo, G. (2005). Reasoning on UML class diagrams. *Artificial intelligence*, *168*(1–2), 70–118. doi:10.1016/j.artint.2005.05.003.

Bonfatti, F., Monari, P.D. & Paganelli, P. (1998). Resource-free and resource-dependent aspects of process modelling: A rule-based conceptual approach. *International Journal of Computer Integrated Manufacturing*, *11*(1), 60–76. doi:10.1080/095119298130985.

Borgo, S., Franssen, M., Garbacz, P., Kitamura, Y., Mizoguchi, R. & Vermaas, P.E. (2014). Technical artifacts: An integrated perspective. *Applied Ontology Journal*, *9*(3–4), 217–235. doi:10.3233/AO-140137.

Borgo, S. & Masolo, C. (2009). Foundational choices in DOLCE. In S. Staab and R. Studer (Eds.), *Handbook on Ontologies* (pp. 361–381). Springer. doi:10.1007/978-3-540-92673-3_16.

Borgo, S., Mizoguchi, R. & Smith, B. (2011). Applied ontology. Special issue on the ontology of functions. *Applied Ontology Journal*, *6*(2), 99–163.

Borgo, S., Sanfilippo, E.M., Šojić, A. & Terkaj, W. (2015). Ontological analysis and engineering standards: An initial study of IFC. In V. Ebrahimipour and S. Yacout (Eds.), *Ontology Modeling in Physical Asset Integrity Management* (pp. 17–43). Cham: Springer.

Borgo, S. & Vieu, L. (2009). Artefacts in formal ontology. In *Philosophy of Technology and Engineering Sciences* (pp. 273–307). Elsevier. doi:10.1016/B978-0-444-51667-1.50015-X.

Casati, R., Varzi, A.C., et al. (1999). *Parts and Places: The Structures of Spatial Representation*. Mit Press.

Castelfranchi, C. (1998). Modelling social action for AI agents. *Artificial intelligence*, *103*(1–2), 157–182.

Chryssolouris, G. (1992). The operation of manufacturing systems. In *Manufacturing Systems: Theory and Practice* (pp. 321–413). New York, NY: Springer New York.

Colledani, M., Terkaj, W., Tolio, T. & Tomasella, M. (2008). Development of a conceptual reference framework to manage manufacturing knowledge related to products, processes and production systems. In A. Bernard and S. Tichkiewitch (Eds.), *Methods and Tools for Effective Knowledge Life-Cycle-Management* (pp. 259–284). Springer. doi:10.1007/978-3-540-78431-9_15.

Dassisti, M., Panetto, H., Tursi, A. & De Nicolò, M. (2008). Ontology-based model for production-control systems interoperability. In *The 5th CIRP Digital Enterprise Technology Conference* (pp. 527–543).

de Farias, T.M., Roxin, A. & Nicolle, C. (2015). IfcWoD, semantically adapting IFC model relations into OWL properties. CoRR. abs/1511.03897. http://arxiv.org/abs/1511.03897.

Erden, M.S., Komoto, H., van Beek, T.J., D'Amelio, V., Echavarria, E. & Tomiyama, T. (2008). A review of function modeling: Approaches and applications. *Ai Edam*, *22*(2), 147–169.

Garbacz, P., Borgo, S., Carrara, M. & Vermaas, P.E. (2011). Two ontology-driven formalisations of functions and their comparison. *Journal of Engineering Design*, *22*(11–12), 733–764. doi:10.1080/09544828.2011.597376.

Grüninger, M. (2009a). The ontological stance for a manufacturing scenario. *Journal of Cases on Information Technology (JCIT)*, *11*(4), 1–25. doi:10.4018/jcit.2009072101.

Grüninger, M. (2009b). Using the PSL ontology. In S. Staab and R. Studer (Eds.), *Handbook on Ontologies* (pp. 423–443). Springer. doi:10.1007/978-3-540-92673-3_19.

Grüninger, M. (n.d.). COLORE. https://github.com/gruninger/colore.

Guarino, N. & Sanfilippo, E.M. (2019). Characterizing IOF terms with the DOLCE and UFO ontologies. In *Proceedings of the Joint Ontology Workshops 2019 (JOWO)*. CEUR Workshop Proceedings (Vol. 2518).

Guizzardi, G., Wagner, G., Almeida, J.P.A. & Guizzardi, R.S. (2015). Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Applied ontology*, *10*(3–4), 259–271. doi:10.3233/AO-150157.

IEC (2013). IEC 62264-1: Enterprise control system integration – part 1: Models and terminology. https://www.iso.org/standard/57308.html.

ISO (2004a). ISO 15531-1: Industrial automation systems and integration – industrial manufacturing management data – part 1: General Overview. https://www.iso.org/standard/28144.html.

ISO (2004b). ISO 15531-31: Industrial automation systems and integration – industrial manufacturing management data – part 31: Resource information model. https://www.iso.org/standard/31583.html.

ISO (2018). ISO 20534: Industrial automation systems and integration – formal semantic models for the configuration of global production networks. https://www.iso.org/standard/68274.html.

Janowicz, K., Haller, A., Cox, S.J.D., Phuoc, D.L. & Lefrançois, M. (2019). SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, *56*, 1–10. doi:10.1016/j.websem.2018.06.003.

Järvenpää, E., Siltala, N., Hylli, O. & Lanz, M. (2019). The development of an ontology for describing the capabilities of manufacturing resources. *Journal of Intelligent Manufacturing*, *30*(2), 959–978. doi:10.1007/s10845-018-1427-6.

Kádár, B., Lengyel, A., Monostori, L., Suginishi, Y., Pfeiffer, A. & Nonaka, Y. (2010). Enhanced control of complex production structures by tight coupling of the digital and the physical worlds. *CIRP Annals*, *59*(1), 437–440. doi:10.1016/j.cirp.2010.03.123.

Kamrani, A.K. & Vijayan, A. (2008). Template-based integrated design: A case study. In A.K. Kamrani and E.S.A. Nasr (Eds.), *Collaborative Engineering: Theory and Practice* (pp. 19–41). Boston, MA: Springer. doi:10.1007/978-0-387-47321-5_2.

Kitamura, Y. & Mizoguchi, R. (2013). Ontological characterization of functions: Perspectives for capturing functions and modeling guidelines. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *27*(3), 259–269. doi:10.1017/S0890060413000267.

Lemaignan, S., Siadat, A., Dantan, J.-Y. & Semenenko, A. (2006). MASON: A proposal for an ontology of manufacturing domain. In *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)* (pp. 195–200). IEEE.

Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Richter, S., Chipman, T., Weise, M. & Wix, J. (2013). Industry foundation classes IFC4 official release. https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC2/HTML/.

Loebe, F. (2007). Abstract vs. social roles – towards a general theoretical account of roles. *Applied Ontology*, *2*(2), 127–158.

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

Masolo, C. & Sanfilippo, E.M. (2020). Technical artefact theories: A comparative study and a new empirical approach. *Review of Philosophy and Psychology*, *11*(4), 831–858. doi:10.1007/s13164-020-00475-9.

Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., Guarino, N., et al. (2004). Social roles and their descriptions. In *KR* (pp. 267–277).

Mizoguchi, R., Galton, A., Kitamura, Y. & Kozaki, K. (2015). Families of roles: A new theory of occurrent-dependent roles. *Appl. Ontology*, *10*(3–4), 367–399. doi:10.3233/AO-150148.

Mizoguchi, R., Kitamura, Y. & Borgo, S. (2016). A unifying definition for artifact and biological functions. *Applied Ontology*, *11*(2), 129–154. doi:10.3233/AO-160165.

Mumford, S. (2003). *Dispositions*. Clarendon Press.

Otte, J.N., Kiritsis, D., Ali, M.M., Yang, R., Zhang, B., Rudnicki, R., Rai, R. & Smith, B. (2019). An ontological approach to representing the product life cycle. *Applied Ontology*, *14*(2), 179–197. doi:10.3233/AO-190210.

Pauwels, P., Krijnen, T., Terkaj, W. & Beetz, J. (2017). Enhancing the ifcOWL ontology with an alternative representation for geometric data. *Automation in Construction*, *80*, 77–94. doi:10.1016/j.autcon.2017.03.001.

Qiao, L., Kao, S. & Zhang, Y. (2011). Manufacturing process modelling using process specification language. *The International Journal of Advanced Manufacturing Technology*, *55*(5–8), 549–563. doi:10.1007/s00170-010-3115-3.

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

Sanfilippo, E.M., Terkaj, W. & Borgo, S. (2019). Resources in manufacturing. In *Proceedings of the Joint Ontology Workshops 2019 (JOWO)*. CEUR Workshop Proceedings (Vol. 2518).

Sarkar, A. & Šormaz, D. (2019). Ontology model for process level capabilities of manufacturing resources. *Procedia Manufacturing*, *39*, 1889–1898. doi:10.1016/j.promfg.2020.01.244.

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

Solano, L., Rosado, P. & Romero, F. (2014). Knowledge representation for product and processes development planning in collaborative environments. *International Journal of Computer Integrated Manufacturing*, *27*(8), 787–801. doi:10.1080/0951192X.2013.834480.

Terkaj, W., Gaboardi, P., Trevisan, C., Tolio, T. & Urgo, M. (2019). A digital factory platform for the design of roll shop plants. *CIRP Journal of Manufacturing Science and Technology*, *26*, 88–93. doi:10.1016/j.cirpj.2019.04.007.

Terkaj, W. & Urgo, M. (2012). Virtual factory data model to support performance evaluation of production systems. In *CEUR Workshop Proceedings* (Vol. 886, pp. 44–58).

Urgo, M. & Terkaj, W. (2020). Formal modelling of release control policies as a plug-in for performance evaluation of manufacturing systems. *CIRP Annals*, *69*(1), 377–380. doi:10.1016/j.cirp.2020.04.007.

Yang, X., Malak, R.C., Lauer, C., Weidig, C., Hagen, H., Hamann, B., Aurich, J.C. & Kreylos, O. (2015). Manufacturing system design with virtual factory tools. *International Journal of Computer Integrated Manufacturing*, *28*(1), 25–40. doi:10.1080/0951192X.2013.800948.