



Ontology-based knowledge representation for additive manufacturing

Emilio Sanfilippo, Farouk Belkadi, Alain Bernard

► To cite this version:

Emilio Sanfilippo, Farouk Belkadi, Alain Bernard. Ontology-based knowledge representation for additive manufacturing. *Computers in Industry*, 2019, 109, pp.182-194. 10.1016/j.compind.2019.03.006 . hal-02277050

HAL Id: hal-02277050

<https://hal.science/hal-02277050>

Submitted on 22 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Ontology-based knowledge representation for additive manufacturing

Emilio M. Sanfilippo*, Farouk Belkadi, and Alain Bernard
Laboratory of Digital Sciences of Nantes,
École Centrale of Nantes,
CNRS UMR 6004

Abstract

The flourishing development of additive manufacturing (AM) technologies calls for robust IT methodologies and solutions to manage the plethora of data that is generated in the AM value chain. The purpose of this paper is to propose a principled knowledge-based model for AM in the form of a computational ontology. As corpus of formally represented knowledge, the ontology constitutes the backbone structure to organize AM data and automatically reason over experts' knowledge for data validation, ultimately supporting the development of algorithms and applications for decision making. By the end of the paper we show some modeling and reasoning examples based on the use of the proposed ontology in a prototype Web application.

Key words: manufacturing, additive manufacturing, knowledge representation, reasoning, ontology, DOLCE

*Corresponding author: emiliosanfilippo@gmail.com (permanent address).

Ontology-based knowledge representation for additive manufacturing

Abstract

The flourishing development of additive manufacturing (AM) technologies calls for robust IT methodologies and solutions to manage the plethora of data that is generated in the AM value chain. The purpose of this paper is to propose a principled knowledge-based model for AM in the form of a computational ontology. As corpus of formally represented knowledge, the ontology constitutes the backbone structure to organize AM data and automatically reason over experts' knowledge for data validation, ultimately supporting the development of algorithms and applications for decision making. By the end of the paper we show some modeling and reasoning examples based on the use of the proposed ontology in a prototype Web application.

Key words: manufacturing, additive manufacturing, knowledge representation, reasoning, ontology, DOLCE

1 Introduction

The development and industrial adoption of additive manufacturing (AM) technologies is bringing about a situation where the entire product development chain is *digitalized*, from the initial design of products to the simulation of processes for the optimization of fabrication, or the maintenance of products via online service systems [31, 35]. Additionally, since AM machines are fully equipped with sensors, a plethora of data about AM processes is nowadays available to companies, which need methodologies and technologies to manage the data and capitalize the experience thereby stored [4, 52].¹

One of the challenges brought about by the organization of the AM value chain is the development of computer systems able to aid decision making but also to support data management in a homogeneous and consistent way [5]. Computer Decision Aid systems play an important role for AM to assist experts in selecting fabrication parameters (among other relevant decisions), whose dependencies and inter-relations need to be carefully evaluated and analyzed to gain full control over fabrication [12, 18]. Concerning data organization,

¹Under the pressure of AM, the discipline of *AM Informatics* is born as “the science of managing AM data across its lifecycle with full maintenance of the complex relationship between the part geometry, material, and individual processes used to create the final part” [35].

different working groups rely on their own models to store and manage data [35]. A consequence is that when multiple communities need to share and possibly integrate data, there is no guarantee for the data to be manipulated in an interoperable manner [6, 37].

A way to both support decision making and tackle data management issues is by means of knowledge-based systems, where experts' knowledge is encoded in formal languages whose semantic is processable by computer systems. Ontologies [22] have proven useful to both exchange data across applications and automatically reason over knowledge and data for knowledge discovery, constraints validation or decision making (see, e.g., [15, 57]).

The purpose of this research work is to propose an ontology for AM that is application and vendor independent to be reused across computer systems to support knowledge and data management in an interoperable manner. Also, since the model formally represents experts' knowledge in a computer amenable way, it can constitute the knowledge base for decision making algorithms to be exploited in AM applications.

The remaining of the paper is structured as follows. The next section introduces the context of our research and addresses the need for an ontology-based modeling approach. The state of art relevant to our work is presented in Section 3. The ontology is presented throughout Section 4, whereas Section 5 exemplifies the use of the ontology in a prototype Web application. Section 6 concludes the paper by addressing some challenges for future research work. Finally, the Appendix shows some queries and reasoning expressed over the knowledge base.

2 Research context

From both the academic literature [21, 54, 59] and real-world industrial practices,² it emerges that the data manipulated in AM tasks is highly heterogeneous in terms of both their originating sources and intended meaning. Figure 1 provides an overall view on the entities involved in the AM value chain. The idea is that an AM machine, which lies at the centre of the figure for its importance, requires and produces a plethora of data, which need to be accessed by different parties and computer applications. In order to make it possible, a knowledge base shared by all applications in the loop is necessary. Since data have to be managed in an *integrated* way, knowledge models able to deal with the representation of entities like processes, products, materials, and machines, including the relations among them, are needed [4, 32, 56]. These models have to be (i) general enough to encompass multiple entities and points of view, (ii) easily extendable according to application requirements, and (iii) vendors-independent to be easily reused and shared. Differently, companies manage data through proprietary legacy systems, which are developed to match very specific requirements [35]. It is not surprising that when multiple stakeholders need to interact by mutually exchanging data, the intended meaning of the data is often lost [37, 38]. Additionally, models for AM need to take into account the

²The reader can refer to [16, 28, 35] for an overview of industrial experiences in AM.

representation of (process, product, machine, etc.) parameters to explicitly analyze and specify their inter-relations and dependencies, as well as to aid experts during decision making procedures [4, 59].

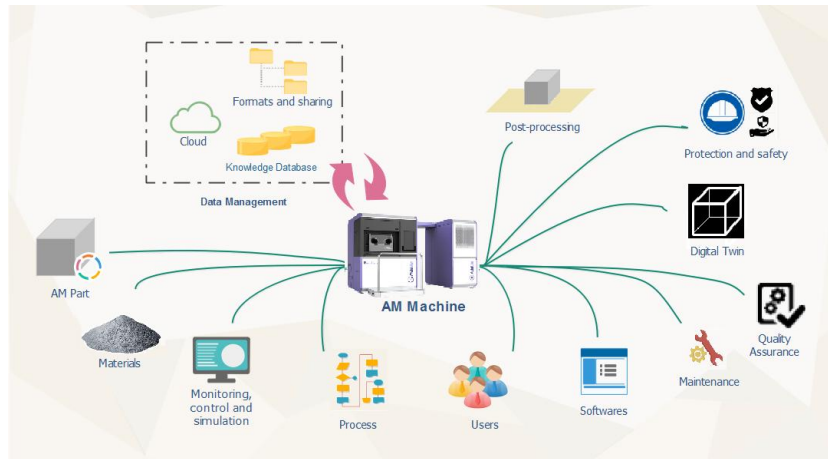


Figure 1: Broad view on the entities involved in the AM value chain

To deal with knowledge representation, data management, and computer aided decision making, we propose the development of computer applications based on knowledge engineering modeling techniques. In particular, the proposal explored in the paper is the development of a knowledge base as reference model to manage AM data and knowledge in a homogeneous and consistent way. Figure 2 exemplifies our approach. By looking at the figure, the knowledge base is the ground element to guarantee the seamless organization of data in a common repository, but also to aid decision making on, e.g., the selection of the parameters that are better suited to fabricate products.

As mentioned in the Introduction, the proposed knowledge base takes the form of a (populated) ontology, i.e., a vocabulary of terms which are formally characterized to be accessible by both computer systems and human agents [22]. When multiple systems rely on the same ontology, the data they handle is semantically described in the same way. Also, since ontologies encode experts' knowledge in formal or even computational logics, they can be used to automatically reason over knowledge and data.

Ontologies are broadly used in the engineering domain, and international activities led by both academia and industry currently aim at creating libraries of reference ontologies for reuse.³ However, because of the novelty of approaches dealing with the organization of the AM value chain, the literature on ontologies for AM is poor in comparison with domains like machining or assembly (see, e.g., [29, 37, 51]). The next section provides a

³The reader can refer to the Industrial Ontology Foundry (IOF) at: <https://www.industrialontologies.org>, last accessed March 2019.

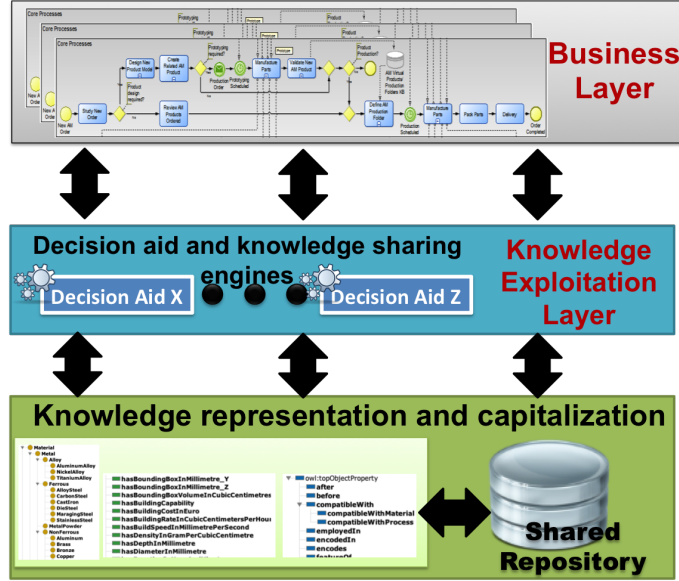


Figure 2: Knowledge-based approach

review of the state of art relevant to our purposes; by identifying the limits of current works, we introduce the ontology in Section 4.

3 Ontologies for additive manufacturing

Researchers at the U.S. NIST have pioneered the use of ontologies for AM applications. In [42, 55] the authors propose an ontology primarily targeted to the data management of powder bed fusion processes. The ontology builds on classes to represent materials, products' components, process parameters, and processes' parts. Unfortunately, the ontology blurs basic ontological modeling principles like the distinction between *subsumption* and *parthood* relations, or between contextual and non-contextual knowledge. For instance, the class for parameters subsumes lasers and sensors. It should be clear that the latter *count as* parameters only within specific application contexts; e.g., a laser is not a parameter for its manufacturer. Modeling pitfalls concerning the erroneous use of subsumption and parthood occur in [13, 32], too, preventing the reuse of the ontologies.

Eddy and colleagues [14] propose an ontology targeted to AM process planning. The ontology covers the ASTM classification of AM processes (cf. [18]), as well as various classes for processes' participants. The ontology has been recently re-engineered [25] according to the upper-level ontology called *Basic Formal Ontology* (BFO) [2] and BFO-compliant

ontologies like the *Relation Ontology*⁴ and the *Information Artifact Ontology* (IAO).⁵ It goes in the same direction the work presented in [1], where an ontology for AM based on the BFO, the IAO, and the *Common Core Ontology* (CCO)⁶, among others, is introduced. The comparison and possibly interoperability with the ontology in [25] is however left by the authors to future work.

Although the better quality of the ontologies in [1, 14, 25] in comparison with the ontologies previously discussed in this section (e.g., [1, 14, 25] properly distinguish between different relation types), we refrain from reusing them because of the ambiguous nature of some modeling elements imported from BFO-compliant ontologies. In particular, the class `INFORMATIONCONTENTENTITY`, which lays at the heart of the IAO, has instances things like design specifications or process plans, which are meant to be *about* some entities in reality [11]. This is not necessarily true in domains like product design or manufacturing; e.g., in a common planning scenario, at the time in which a process plan is created, there is not the corresponding process, which may never occur if the plan is never realized.⁷

Finally, Qi and colleagues [40] present a preliminary study based on category theory to model the inter-relations between AM parameters, e.g., between layer thickness and particle size distribution. However, because of the adopted formalism, it remains unclear how the ontology can be exploited for application purposes, e.g., to manage or validate stakeholders' data against formally encoded knowledge.

Table 1 provides a summarizing comparison of the state of art in order to better frame the scope of our research. The columns *Product*, *Process*, *Resource*, and *Parameters* are used to assess whether the ontologies cover modeling elements about those entities. As it can be seen from the table, the ASTM classification of AM processes is covered only by some ontologies. Also, the representation of parameters greatly varies across the papers depending on application requirements. The *Reused ontologies* and *Modeling language* columns are self-explicatory; Semantic Web languages like the Web Ontology Language (OWL) [19] and SWRL [27] are mostly used.⁸

The last row of the table introduces the work presented in the following sections. Similarly to (some of the) existing works, the ontology covers the representation of products, processes, and resources. Also, to support a wide modeling of parameters, we rely on the scientific literature on AM, the expertise of academics and industrial stakeholders, as well as AM machines, products, and software vendors' catalogues. From an ontology engineering perspective, on the one hand, differently from [14, 32, 40, 42, 55], the ontology is founded on rigorous modeling principles like the ones discussed in [23, 24]. On the other

⁴<http://www.obofoundry.org/ontology/ro.html>, last accessed March 2019.

⁵<http://www.obofoundry.org/ontology/iao.html>, last accessed March 2019.

⁶<https://www.cubrc.org/index.php/data-science-and-information-fusion/ontology>, last accessed March 2019.

⁷Similar concerns about the IAO have been raised at least in [46, 48].

⁸Since most of the OWL ontologies presented in this section are not available for review or reuse, the table is mainly based on the materials presented in the papers.

Paper	Product	Process	Resource	Parameters	Reused ontologies	Modeling language
[1]	Yes	Yes (ASTM)	Yes	Yes	BFO, IAO, CCO	OWL
[13]	Yes	Yes (no ASTM)	Yes	Yes	None	OWL, SWRL
[14]	Yes	Yes (ASTM)	Yes	Yes	MSDL [43]	OWL, SWRL
[25]	Yes	Yes (no ASTM)	Yes	Yes	BFO, RO, IAO, FBO [17], [14]	OWL, SWRL
[32]	Yes	Yes (no ASTM)	Yes	Yes	None	IDEF0, UML
[40]	No	No	Yes	Yes	None	Category theory
[42, 55]	Yes	Yes (no ASTM)	Yes	Yes	None	OWL
Our work	Yes	Yes (ASTM)	Yes	Yes	DOLCE	OWL, SWRL

Table 1: Comparison of ontologies for additive manufacturing

hand, the ontology trades-off generality with domain-specificity in such a way to cover a broad range of concepts to deal with the heterogeneity of AM knowledge and data, as well as to be easily extended to specific application scenarios. For this purpose, differently from [1, 25], the ontology is based on the *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) [9, 34] as core methodology for knowledge organization, classification, and conceptual analysis. We rely on DOLCE, first, because it covers (high-level) modeling elements which are needed for knowledge representation in manufacturing, e.g., the distinction between objects, processes, and qualities (see Section 4), but also relations like *parthood* or *participation* that are necessary to model the structure of complex items like assemblies and the links between objects and processes, respectively. Second, because DOLCE has been already used for modeling engineering design and manufacturing knowledge (e.g., [7, 8, 50]). Third, because DOLCE is one of the upper-level ontologies guiding the development of a knowledge-based architecture for the Industrial Ontology Foundry (IOF) (cf. Section 2). Therefore, by relying on DOLCE, we support and facilitate the comparison and interoperability with IOF modeling efforts. Since DOLCE is formalized in first-order (modal) logic, we use it as reference framework, whereas the OWL ontology presented in Section 4 simplifies and adapts it to the expressivity of Semantic Web languages.

4 The proposed ontology for additive manufacturing

For limits of space the following sections introduce only some modeling elements of the ontology; the reader can refer to the ontology file for an insightful overview of the ontology’s structure.⁹ Section 4.1 presents the upper-level branch of the ontology, whereas the AM branch is introduced in Section 4.2.

The ontology is developed in OWL by means of Protégé¹⁰. For the sake of readability, we use the syntax of Description Logics [3] throughout the paper. The languages SWRL and SPARQL are used for expressing rules and queries, respectively.¹¹

4.1 The upper-level branch of the ontology

One of the core purposes of upper-level ontologies is to facilitate the grouping of classes sharing common high-level characteristics, e.g., entities that are primarily extended in space like machines, materials, or manufacturing tools, and entities that unfold through time such as manufacturing or maintenance processes. Therefore, by using upper-level classes and relations one avoids the duplication of axioms across a domain-specific ontology. Also,

⁹The ontology is available at [link to be added after review is finalized].

¹⁰<https://protege.stanford.edu/>, last accessed February 2019.

¹¹We prefix axioms with *Ax*, definitions with *Def*, formulas for examples with *f*, rules with *R*, and queries with *Q*. Also, we omit question marks from SWRL variables and PREFIX statements from SPARQL queries.

when multiple application systems are aligned to the same upper-level, they have a higher chance of interoperating for the fact of sharing the same core elements [30].

The most general classes of our ontology, which are all disjoint from each other, are shown in Figure 3.

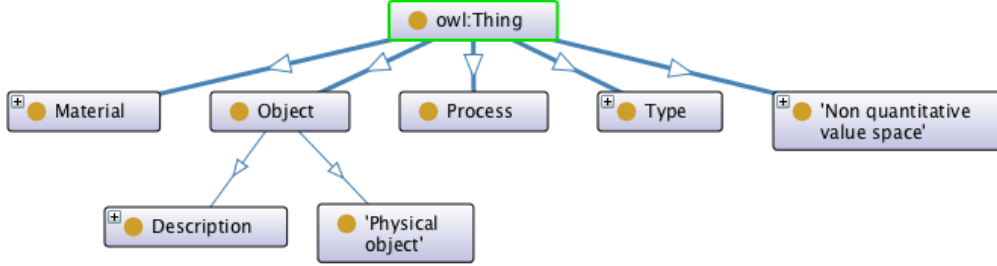


Figure 3: Upper-level taxonomy

Following DOLCE, objects¹² are entities that can change through time while retaining their identities, e.g., a manufacturing tool which remains the same entity (indeed, it keeps the same product ID) although the changes it undergoes due to wear. Examples of objects are AM machines and tools, but also entire factories, simple and assembled products, persons, design specifications, and manufacturing plans.

In order to organize the spectrum of entities classified by OBJECT, this class is specialized in PHYSICALOBJECT and DESCRIPTION (see Figure 3). The first class models objects having a spatial location, e.g., persons, tools, or products. Physical objects are particularly relevant for manufacturing modeling; hence, PHYSICALOBJECT is further specialized in various classes (cf. Section 4.2). The second class groups entities like plans and design specifications from the perspective of their ‘content’. For example, when one creates a CAD model saved in a STL file and duplicates the file, the result is two copies of exactly the same model. It is this latter entity that we call *description* (cf. [36] for a similar approach).¹³

PROCESS is the most general class for entities that unfold through time.¹⁴ Following DOLCE, objects can take part in processes in various ways; *participation* is the most general link holding between objects and processes. We will see in the next section how both PROCESS and *participation* are extended.

¹²We use ‘object’ to replace DOLCE ‘endurant’ in order to use a terminology closer to manufacturing.

¹³In order to distinguish a description from its (possibly multiple) supports (e.g., digital files), PHYSICALOBJECT covers the class DESCRIPTIONSUPPORT. According to the ontology engineering literature (e.g., [53]) digital files are physical objects since they are stored in the physical memory cells of hardwares, hence they have spatial locations. The ontology we present can be consistently extended to model the distinction between digital and non-digital physical objects, if needed by application requirements.

¹⁴DOLCE uses the term ‘perdurant’ whereas we use ‘process’ because this is a more common term in manufacturing [50].

The class MATERIAL¹⁵ is the most general classifier for amounts of matter, e.g., the materials used to fabricate products by AM processes. With this class one can easily model the distinction between *material* and *immaterial* objects. Example of the latter are hole or pocket features [45].

The use of the class NONQUANTITATIVEVALUESPACE has to be contextualized within a global approach to model what DOLCE calls *qualities*, i.e., the characteristics of objects, processes, or materials, e.g., weight, height, diameter, speed, material state, etc. In ontology engineering, various approaches have been proposed for representing qualities, e.g., [24, 33, 41]. For our purposes, we trade-off conceptual expressivity with application concerns; hence, in order to distinguish between qualities and their values, to make sense that qualities' values can be provided according to (different) measurement units, but also to facilitate the creation of data models, we model qualities via relationships rather than classes. In particular, by taking benefit of the OWL distinction between *data properties* and *object properties*, we distinguish between *quantitative* and *non-quantitative* quality-relations, namely, (i) relations linking instances of OBJECT, PROCESS, MATERIAL, or TYPE (see below) to numerical values (e.g., *being 30kg heavy* or *being 3mm height*), and (ii) relations linking instances of these classes to non-numerical values (e.g., *being light*, *being flake-shaped*, *being small*, etc.). The class NONQUANTITATIVEVALUESPACE is used to collect the latter values. E.g., the data property HASTHICKNESSINMILLIMETRE can be used to model the numerical value of a product's layer thickness, whereas the object property HASSHAPEVALUE can be used to model, e.g., the flake shape of metal powders.¹⁶

Finally, the class TYPE is used when quantification over properties is needed within the expressivity of languages based on first-order logic like OWL [9, 47].¹⁷ For manufacturing knowledge representation, this approach is common, e.g., for process planning, when one needs to model the type of machine that is required to be used without however specifying the individual machine that will be possibly used [44].¹⁸ Similarly, in the context of design, reference to types is necessary to talk about the product types under design, whose physical realizations do not necessary exist at the designing time [47]. In the ontology, the relation ISOFTYPE binds an instance of MATERIAL, OBJECT, or PROCESS to a type; e.g., one can say that the material constituting a certain product is of type NICKELALLOY718 (cf. next section).

Table 2 reports the relations used in the upper-level branch of the ontology.

¹⁵In DOLCE this class is called 'amount of matter'.

¹⁶Quantitative and non-quantitative values can be easily related. E.g., one may formally establish that layers qualify as *small* when their thickness is less than 0.2mm.

¹⁷There exists various theories of types in ontology engineering (e.g., [26]). Types with a cognitive dimension can be represented in DOLCE (Core) through the CONCEPT class [9].

¹⁸In the context of process modeling, the reader can refer to the distinction between ACTIVITY (type) and ACTIVITYOCCURRENCE in the Process Specification Language (ISO 18629) [20].

Relation	Domain	Range	Example
AFTER (inverse, BEFORE)	PROCESS	PROCESS	Process p_1 occurs after process p_2
ENCODES (ENCODEDIN)	PHYSICALOBJECT	DESCRIPTION	Description support d encodes CAD model c
HASNON QUANTITATIVEVALUE	OBJECT, MATERIAL, PROCESS	NONQUANTITATIVE-VALUESPACE	Material m has non-quantitative value <i>solid</i>
MADEOF	PHYSICALOBJECT	MATERIAL	Physical object pob is made of material m
PARTICIPATESIN (HASPARTICIPANT)	OBJECT, MATERIAL, TYPE	PROCESS	Object o participates in process p_1
PARTOF (HASPART)	OBJECT, MATERIAL, PROCESS, TYPE, QUALITATIVE-VALUESPACE	OBJECT, MATERIAL, PROCESS, TYPE, NONQUANTITATIVEVALUESPACE	Object o_1 has parts objects o_2 and o_2
ISOFTYPE	OBJECT, MATERIAL, PROCESS	TYPE	Material m is of type NICKELALLOY718

Table 2: Upper-level relations

4.2 The additive manufacturing branch of the ontology

This section shows the extension of the upper-level branch to additive manufacturing. The ontology includes broad classes and relations, which can be easily specialized to meet specific modeling scenarios and requirements.

Machines, products, and features. The class `PHYSICALOBJECT` is extended in two main classes, namely, `PERSON` and `PHYSICALARTEFACT`. The first one is useful to model human agents, e.g., taking part in manufacturing processes, whereas the second one is the most general classifier for physical objects that are created to satisfy market requirements. The two classes are therefore disjoint, since they cannot have common instances.

`PERSON` is specialized to cover roles that human agents may play in engineering contexts. Examples are `MACHINEOPERATOR` and `MFGENGINEER`,¹⁹ among others.

`PHYSICALARTEFACT` covers various classes, among which `MFGDEVICE`, `PRODUCT`, and `FEATURE` are particularly interesting from a manufacturing modeling stance. `MFGDEVICE` is a general classifier to better organize subclasses of `PHYSICALARTEFACT`, e.g., to explicitly state disjointness axioms with other classes. Its instances are made of material and may be adopted in manufacturing processes; examples are machines and their components (e.g., sensors, tools, engines, energy sources, etc.). `PRODUCT` – as the label suggests – models physical goods that are fabricated to bring economical value to manufacturing companies. `FEATURE` refers to things like holes, steps, bumps or protrusions, among others [45]. This class is disjoint from both `MFGDEVICE` and `PRODUCT`.

`MFGDEVICE` is specialized in various classes, the most important one being `MFGMACHINE` for manufacturing machines. `AMMACHINE` specializes the latter class for the explicit modeling of additive manufacturing machines, see (Ax1) and (Ax2). The relationship `MECHANISMOF` used in (Ax1)–(Ax2) is taken from previous work based on both `DOLCE` and `IDEF` [44]. It specializes the relationship of *participation* (cf. Table 2) and grasps the link between manufacturing processes and their ‘active’ participants.²⁰ By axioms (Ax1)–(Ax2), machines do not necessarily participate in manufacturing processes. The reason is that an individual machine may be never employed in any manufacturing process.

Ax1 $\text{MFGMACHINE} \sqsubseteq \text{MFGDEVICE} \sqcap \forall \text{MECHANISMOF.MFGPROCESS}$

Ax2 $\text{AMMACHINE} \sqsubseteq \text{MFGMACHINE} \sqcap \forall \text{MECHANISMOF.AMPROCESS}$

Classes like `ENERGYSOURCE` (e.g., lasers), `BUILDPLATFORM` or `BUILDCHAMBER` are included in the taxonomy of `PHYSICALARTEFACT` and are useful to model (some of) the components making up whole AM machines.²¹

¹⁹We use the acronyms *Mfg* and *Am* to label some classes, the former standing for manufacturing, the latter for additive manufacturing. Also, the modeling of roles is based on [23, 24].

²⁰The representation of manufacturing processes is discussed at the end of the section.

²¹Our purpose is not to dig into the representation of AM machines. The reason for having classes like `ENERGYSOURCE` or `BUILDPLATFORM` is to show how they can be accommodated within the ontology.

In order to express the link between an assembled physical artefact and its components, the `HASCOMPONENT` relation is introduced. It specializes `HASPART` and holds only between physical artefacts. The class `MFGMACHINECOMPONENT` is introduced in (Def1).

Def1 $\text{MFGMACHINECOMPONENT} \equiv \text{MFGDEVICE} \sqcap \exists \text{COMPONENTOF.MFGMACHINE}$

Classes like `BUILDCHAMBER` or `BUILDPLATFORM` are not subclasses of `MFGMACHINECOMPONENT`, since it is not necessarily the case that their instances are components of machines. Consider, e.g., an instance of `BUILDPLATFORM`, call it *bp*, which at a time *t* is not related to any machine and cannot be therefore classified as machine component. However, when *bp* relates at time *t'* different from *t* to an individual machine via `HASCOMPONENT`, it is automatically classified as machine component; cf. the (simplified) example in formulas (f1)–(f4), where the latter formula is the result of the reasoning.

f1 `BUILDPLATFORM(bp)`

f2 `MFGMACHINE(mch)`

f3 `HASCOMPONENT(mch, bp)`

f4 `MFGMACHINECOMPONENT(bp)`

`PRODUCT` is extended in `AMPRODUCT` to explicitly cover products resulting from AM processes (Ax3). In axiom (Ax3), the relationship `OUTPUTOF`, based on DOLCE and IDEF [44], specializes *participation* to grasp the outcomes of manufacturing processes.

Ax3 $\text{AMPRODUCT} \sqsubseteq \text{PRODUCT} \sqcap \exists \text{OUTPUTOF.AMPROCESS}$

Following AM experts' knowledge [18], it is relevant to characterize the distinction between products and AM products, since only the latter are fabricated by the addition of multiple layers of material. An AM product is therefore understood as the (ordered) sum of layers fabricated during an AM process. In the ontology this is (partially) grasped by (Ax4), where `HASLAYER` captures the link between AM products and layers.

Ax4 $\text{AMPRODUCT} \sqsubseteq \exists \text{HASLAYER.AMLAYER}$

Note that, first, layers are physical artefacts resulting from AM processes; they are part of AM products but are not products on their own. Hence, `AMPRODUCT` and `AMLAYER` are disjoint classes. Second, layers are not components of AM products, hence the relation `HASCOMPONENT` cannot be used to link a layer to a product.

As mentioned at the beginning of the section, the representation of physical artefacts includes a taxonomy of feature classes, which are of key relevance for knowledge representation and data management in AM [49, 58]. Recall that there is no exhaustive list of features [45], hence the ontology covers only some common feature classes, which can be extended to meet specific application requirements.

The relationship `FEATUREOF` is used to relate instances of `FEATURE` to instances of either `PRODUCT` or `AMLAYER`. Also, (R1) establishes that when a feature is made of material and is related via `FEATUREOF` to a product, then it is (proper) part of the product. For example, a rib is part of the product to which it is related. Clearly, this does not imply that features are products' components, which is formally excluded in the ontology.

$$\mathbf{R1} \quad \text{FEATUREOF}(x, y) \wedge \text{MADEOF}(x, z) \rightarrow \text{MADEOF}(y, z) \wedge \text{PARTOF}(x, y)$$

Figure 4 shows a partial view on the taxonomy of `PHYSICALARTEFACT`. The reader can refer to the OWL file for a detailed view on the ontology.

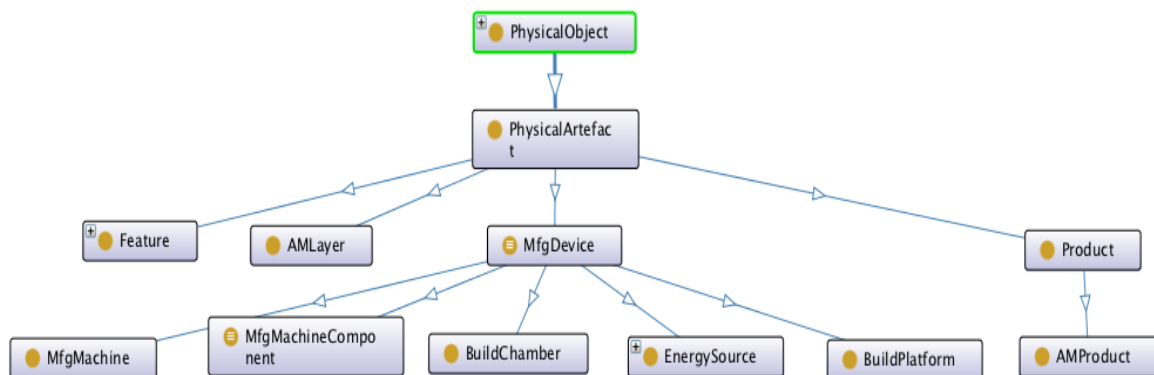


Figure 4: Taxonomy of `PHYSICALARTEFACT` (partial view)

Manufacturing types. The `TYPE` class is extended to cover various classes, including `MATERIALTYPE`, `MACHINETYPE`, and `PROCESSTYPE` (see below). With respect to what said in the previous section, the reader should pay attention in distinguishing types from the spatio-temporal individuals that possibly satisfy them, e.g., `MATERIAL` *vs* `MATERIALTYPE`: instances of the former are the specific amounts of matter that constitute individual objects, whereas instances of the latter are used to model the types they exemplify (e.g., maraging steel 18 Ni 300, protogen 18420).

As previously said, modeling types in manufacturing ontologies is useful, especially in design or planning contexts, when experts need to refer to certain entities without reference to their physical counterparts, e.g., because the latter do not exist at the designing (or planning) time. E.g., machine types are the sorts of entities described in machines vendors' catalogues. Indeed, when one reads that a machine is compatible with certain materials, it is clear that reference is not done to specific physical machines (or materials) but to

non-physical entities that are only *possibly* realized in physical machines. In the proposed ontology this non-physical entity corresponds to a machine type defined by multiple properties such as building volume, building capabilities, etc. (cf. [10] for a similar approach). For our purposes, types are particularly useful to model properties related to the compliance of machine types with material types, a restriction that affects the participation of both (physical) machines and materials in individual processes. Rule (R2) guarantees, indeed, that when a material is the *input* of a process, which uses a certain machine as mechanism, the machine type exemplified by the machine has to be compatible with the material type exemplified by the material (INPUTOF is a further extension of *participation* based on DOLCE and IDEF [44]).²²

R2 $\text{MACHINE}(x) \wedge \text{ISOFTYPE}(x, y) \wedge \text{PROCESS}(z) \wedge \text{MATERIAL}(v) \wedge \text{ISOFTYPE}(v, w) \wedge \text{MECHANISMOF}(x, z) \wedge \text{INPUTOF}(v, z) \rightarrow \text{COMPATIBLEWITHMATERIALTYPE}(y, w)$

For example, consider the AM machine type called FORMUP TYPE, which – according to its vendor’s catalogue – is compatible only with the metal types specified by (f5).

f5 $\text{FORMUP TYPE} \sqsubseteq \text{MACHINETYPE} \sqcap \forall \text{COMPATIBLEWITHMATERIALTYPE} . (\text{ALUMINIUMALLOYALSi10Mg} \sqcup \text{TITANIUMALLOYTi6A4V} \sqcup \text{MARAGINGSTEEL18Ni300TYPE} \sqcup \text{NICKELALLOY625TYPE} \sqcup \text{NICKELALLOY718TYPE} \sqcup \text{STAINLESSSTEEL316LTYPE})$

Assume that an individual physical machine of type FormUp is used as mechanism in a process having input a material of polymer type. Stating this fact in the ontology raises an inconsistency because of (R2) and (f5). The next section shows how these sorts of constraints can be useful to validate additive manufacturing data against experts’ knowledge.

AM processes. The general class for the representation of processes (PROCESS) is extended in MFGPROCESS and AMPROCESS to cover manufacturing and AM processes, respectively; a similar distinction is done at the type level. Coherently with the overall approach to represent types, the class AMPROCESSTYPE is specialized according to the ASTM classification (see [18]) into seven main subclasses, namely, BINDERJETTINGPROCESSTYPE, DIRECTEDENERGYDEPOSITIONPROCESSTYPE, MATERIAL EXTRUSIONPROCESSTYPE, MATERIALJETTINGPROCESSTYPE, SHEETLAMINATIONPROCESSTYPE, VATPHOTOPOLYMERISATIONPROCESSTYPE, and POWDERBEDFUSIONPROCESSTYPE. For the sake of the example, the latter class is specialized into DIRECTMETALLASERSINTERINGTYPE, ELECTRONBEAMMELTINGTYPE, SELECTIVEHEATSINTERINGTYPE, SELECTIVELASERMELTINGTYPE, and SELECTIVELASERSINTERINGTYPE.²³

²²Differently from MECHANISMOF, INPUTOF expresses a form of ‘passive’ participation in a process, e.g., a material ‘undergoing’ (transformed by) a manufacturing process.

²³We already introduced various relationships to link objects or materials to processes.

Finally, the ontology covers various parameters, mostly modeled as OWL data properties, to represent, e.g., the maximum power value of lasers, the velocity of scanning processes, hatch distances, the building volume capabilities of machines, and so on. The reader can refer to the OWL file for an insightful overview of these relations.

Figure 5 shows some of the classes introduced across Section 4.2.

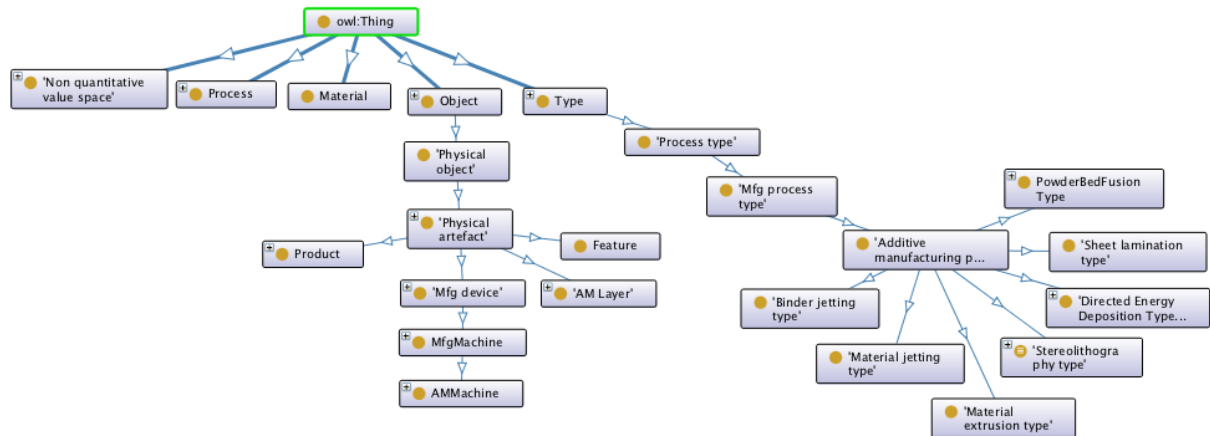


Figure 5: The AM ontology branch (partial view)

5 A prototypical ontology-based Web application for additive manufacturing

This section shows the use of the ontology in a Web application, which is developed in tandem with industrial partners involved in a research project as prototype module within a larger information system. The application is currently used in experimental settings to populate the ontology and guarantee that it matches with both experts' knowledge and stakeholders' requirements.

The main purposes of the application are (i) to support the specification of what the domain experts involved in the project call *templates*; (ii) to (re-)use the templates for the definition of *cases*, (iii) to validate both templates and cases, and (iv) to facilitate their reuse in new AM projects. Figure 6 provides a schematic view of the application, which is formed by a user interface mainly used to take users' inputs, and a server-side module that, among other things, allows to manipulate data inputs, validate them against the formal constraints of the knowledge base, store and retrieve them for reuse.

A template is a collection of AM data; for instance, a *product template* is used to specify the nominal characteristics of a desired product like its dimensions, material type or

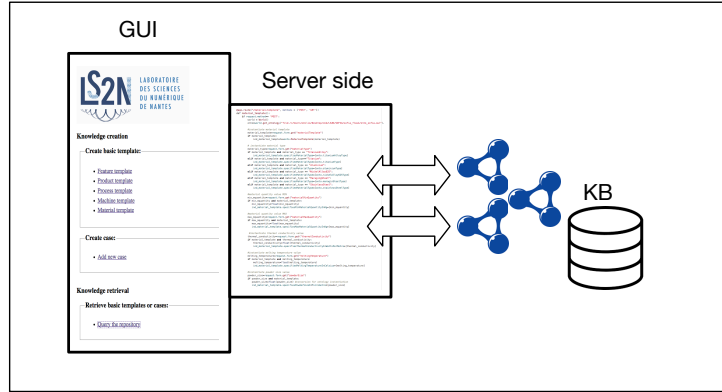


Figure 6: Schema of the Web application

features; or, a *machine template* is used to introduce in the knowledge base a machine type along with its characterizing properties, e.g., building volume capabilities, compatibility with material types, energy source’s power, the capability to produce features of certain dimensions, etc. Along with product and machine templates, the application covers *process*, *material*, and *feature templates*. A *case* is a collection of templates which are needed for the additive manufacturing of a product. It should be clear that the core motivation to distinguish templates from cases is that the former can be reused and customized for multiple cases.

From the perspective of the ontology, product, process, feature templates, and cases are modeled through two main classes, `TEMPLATE` and `CASE`, respectively, both of which specialize the more general `DESCRIPTION` class (see Figure 3). The relation `HAS_TEMPLATE` specializes `HAS_PART` and links cases to templates. Material and machine templates are modeled through the classes `MATERIAL_TYPE` and `MACHINE_TYPE`, the reason being that these templates are used to insert new types in the ontology, e.g., a new machine type and the material types that it is able to process. As we will see in the example below, the relationships `SPECIFIES_MATERIAL_TYPE` and `SPECIFIES_MACHINE_TYPE` can be used to relate a template (or case) to material and machine types, respectively.²⁴

The content of each template in the Web application is structured according to manufacturing knowledge provided by the academic literature, by machines and materials vendors’ catalogs, and by the domain experts involved in the project. Since the ontology formally specifies the overall body of knowledge, all data field entries in the application are modeled through the elements and constraints of the ontology. For instance, a process template allows to define machine and fabrication parameters such as build orientation, layer thickness, hatch distance, scanning speed, laser mode, laser power, laser beam spot diameter, and

²⁴Relations starting with the term *specifies* have `DESCRIPTION` as domain.

laser pulse duration, among others, where each parameter is mapped to an element of the ontology. E.g., assuming that scanning speed is measured in millimetres per second (mm/s), the ontology covers the data property relations `SPECIFIESMINSCAN SPEED IN MILLIMETRES PER SECOND` and `SPECIFIESMAXSCAN SPEED IN MILLIMETRES PER SECOND` for the minima and maxima scanning speed values, respectively.²⁵

For the sake of the example, assume we need to specify a case for manufacturing solid coupons made of nickel alloy 625 with dimensions of 16x16x15 mm by using a machine of type FormUp.²⁶ According to our application pipeline, we first define the required templates, which are then reused in a single case. Figure 7 shows the machine and product templates; as it can be seen from the figure, users can specify various data fields, but they also have the possibility of customizing templates by adding new fields. Templates may be fully or only partially filled; e.g., by looking at Figure 7, the fields about features are not filled, because they are not necessary for the scenario at hand. Once a template is saved, the data is pushed into the knowledge base to populate the ontology; Figure 8 shows the product template inserted into the knowledge base. As said, each data field in the Web application corresponds to a modeling element of the ontology; e.g., from Figure 8 it can be seen the use of the relations `SPECIFIESMINHEIGHT IN MILLIMETRE` and `SPECIFIESMAXHEIGHT IN MILLIMETRE` to instantiate the product’s height values provided through the data fields *Product length (in mm)* in the application.

Assuming that we have already inserted a process template in the knowledge base (see Figure 9), a case can be specified to organize together the templates, refine them – when needed – and validate their dependencies. As an example, consider a modeling situation in which the user, when specifying the case, changes the material type previously selected in the product template from nickel alloy 625 to stainless steel CX. Once the user tries to submit the case to the knowledge base, the application triggers an error message, since FormUp machines cannot be used with this material type (recall formula (f5)); therefore the user has to revise the inserted data in order to proceed. Similar constraints are implemented, e.g., to guarantee that the range of selected parameters such as laser power or scanning speed are configurable on machines of the selected type, or that cases and templates have to consistently model the inter-relations between machine and material types. For instance, (R3) establishes that a machine type cannot be configured with a laser power greater than a predefined maximum value. Should the rule be triggered, the knowledge base becomes inconsistent and the user needs to revise the specified data.²⁷ In this way we guarantee that the data specified through the Web application is consistent with the knowledge formalized in the ontology (see the Appendix section for examples of reasonings for constraints checks). Also, the formal constraints in the ontology can be used in tandem with scenario-based rules to support experts’ decisions making. For the sake of

²⁵All data properties for parameters with numerical values cover both minima and maxima values.

²⁶The example is based on the data provided in [12], which have been adapted to our purposes.

²⁷In order to avoid making the knowledge base inconsistent, an alternative approach can be adopted to infer that the selected machine type cannot be used for that specific purpose.

Fill out the machine template:

Machine characteristics:

Select machine type:

Produced by:

Maximum power consumption (in kW):

Machine building volume capability (in mm):

- Max value in X:
- Max value in Y:
- Max value in Z:

Compatible with material type:

- ☐ Titanium alloy Ti6A4V
- ☒ Aluminium alloy AlSi10Mg
- ☐ Aluminium alloy AlSi112
- ☒ Nickel alloy 625
- ☒ Nickel alloy 718
- ☐ Nickel alloy HX
- ☐ Maraging steel 18Ni300
- ☐ Stainless steel 316L
- ☐ Stainless steel 17-4 PH
- ☐ StainlessSteel CX

Laser type:

- ☒ Yb fiber laser
- ☐ CO₂ laser
- ☐ YAG laser

Laser power (in W):

- Max value:

Fill out the product template:

General data:

Template name:

Key characteristics

Surface roughness (ISO 1302:1992): Roughness value (in µm):

Made of material type:

Product length (in mm):

- Min value:
- Max value:

Product width (in mm):

- Min value:
- Max value:

Product height (in mm):

- Min value:
- Max value:

Feature characteristics

Feature template:

Feature type:

Form feature dimensions (in mm):

- Min height value:
- Max height value:
- Min width value:
- Max width value:
- Min length value:
- Max length value:

Figure 7: Machine and product templates

example, by rule (R4) a reasoner is able to infer the laser power parameter to be configured on the machine of the preselected FormUp type. It should be clear that, on the one hand, these sorts of rules for decision making can be useful to (semi-) automatise decision making processes especially when multiple dependencies between data have to be checked. On the other hand, these rules are context-dependent and efforts are required to formalize experts' knowledge in computational terms.

R3 $\text{DESCRIPTION}(x) \wedge \text{SPECIFIESMACHINETYPE}(x, y) \wedge$
 $\text{HASMAXLASERPOWERINWATT}(y, v) \wedge \text{SPECIFIESMAXLASERPOWERINWATT}(x, z) \wedge$
 $\text{GREATERTHAN}(z, v) \rightarrow \text{owl : Nothing}(x)$

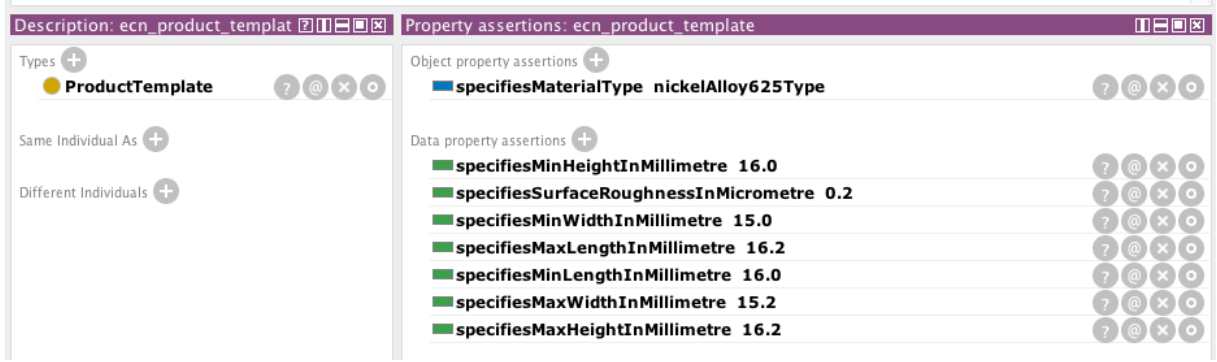


Figure 8: Example of product template added to the knowledge base

R4 $\text{DESCRIPTION}(x) \wedge$
 $\text{SPECIFIESMACHINETYPE}(x, \text{formUpType}) \wedge$
 $\text{SPECIFIESMATERIALTYPE}(x, \text{nickelAlloy625Type}) \wedge$
 $\text{SPECIFIESMAXSCAN SPEED IN MILLIMETRE PER SECOND}(x, 875) \wedge$
 $\text{SPECIFIESHATCHDISTANCE IN MILLIMETRE}(x, 0.10) \wedge$
 $\text{SPECIFIESLAYER THICKNESS IN MICROMETRE}(x, 20) \rightarrow$
 $\text{SPECIFIESMAXLASERPOWER IN WATTS}(x, 169)$

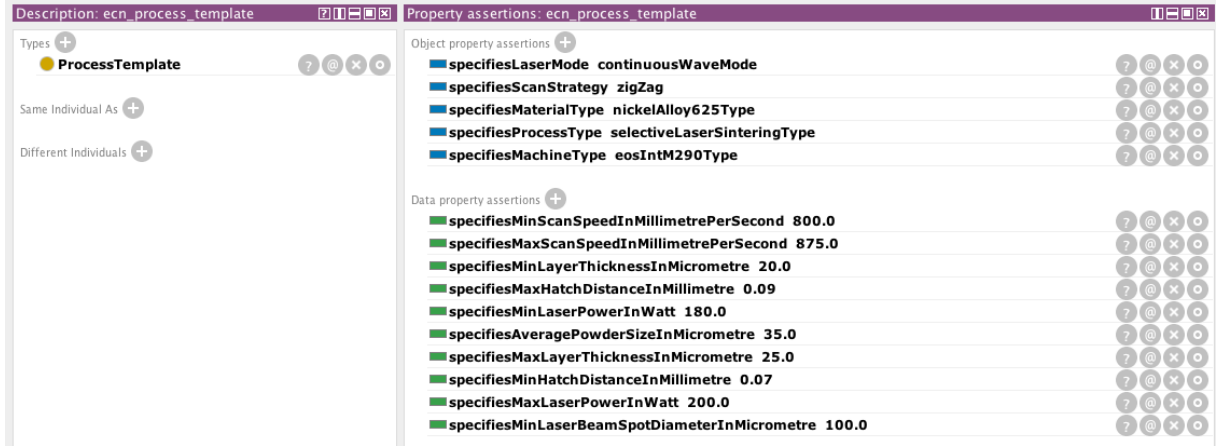


Figure 9: Example of process template added to the knowledge base

Once the knowledge base is populated with users inputs, queries can be formulated to retrieve data for reuse. In the prototype of the application, SPARQL [39] queries can be

expressed by filling out forms similar to the ones showed in Figure 7. For instance, let us assume that the user wishes to retrieve data concerning the compatibility between machine and material types. By using the Web application, the user selects the desired values; the selection is converted in a SPARQL query, like the one showed in (Q1), and data – if found – is showed (see Table 3). The reader can refer to the Appendix for further query examples.

Q1 For each machine type in the KB, retrieve the compatible material types.

```
SELECT ?machineType ?materialType WHERE
  {?machineType a MachineType;
   compatibleWithMaterialType ?materialType.}
```

machineType	materialType
proXDMP200Type	stainlessSteel174PHType
	aluminiumAlloyAlSi12Type
eosIntM290Type	nickelAlloy718Type
	nickelAlloy625Type
	aluminiumAlloyAlSi10MgType
formUpType	aluminiumAlloyAlSi10MgType
	nickelAlloy718Type
	nickelAlloy625Type
	titaniumAlloyTi6A4VType
	stainlessSteel316LType
	maragingSteel18Ni300Type

Table 3: Result of (Q1)

6 Conclusion

The paper presents a computational ontology to represent and reason over AM knowledge and data. The development of the ontology has been motivated with respect to the need for a reference knowledge model to harmonize data management, support data sharing, and validate data against experts’ knowledge. Also, the ontology constitutes the semantic backbone of the information system architecture developed within a research project. As part of this architecture, we presented a prototype Web application which is used to create, validate, and reuse AM data and knowledge. By relying on the ontology, the data managed through the application is indeed semantically characterized to facilitate their sharing across multiple stakeholders. Additionally, by reasoning over the ontology, the data can be checked against possible pitfall.

With respect to the state of art about ontologies for AM, our approach relies on both experts’ knowledge and well-known modeling principles. The purpose is, first, to rely on rigorous modeling criteria; second, to facilitate the reuse of the ontology and its extension for specific application requirements. As discussed throughout Section 3, existing AM ontologies are not directly reused mainly because the modeling patterns upon which they rely are either mistaken or not well suited for manufacturing knowledge representation. Our research work is therefore a contribution towards a principled knowledge base for AM.

Since the additive manufacturing domain is constantly evolving under the pressure of both academia and industry, it is not surprising that further work on our proposal is required. In the next future we expect to use the ontology to develop algorithms and softwares to support data sharing and data tracing along the entire AM value chain, as well as to aid experts’ decision making. Concerning this latter application scenario, at the current state of the Web application, data are either manually inserted or reused by retrieving them from existing templates or cases. In the future version of the application, templates and cases will be (semi-) automatically filled by algorithmically matching users’ requirements and Key Performance Indicators (KPI) to both experts’ knowledge and the past experience stored in the knowledge base. As shown in Section 5, formal rules can be already used to support decision making processes but in order to foster the use of the application in real-world settings, the presented approach needs to be complemented with techniques such as multi-attributes case-based reasoning or concepts similarity measures, among others. To achieve these results, the ontology will be continuously maintained by populating it with research and stakeholders data but also by updating its structure whenever this will be needed. Finally, in order to share our research results with the ontology engineering community active in the manufacturing domain, the ontology will be presented to the Industrial Foundry Ontology (IOF) initiative in such a way to facilitate its reuse and adoption.

References

- [1] ALI, M. M., RAI, R., OTTE, J. N., AND SMITH, B. A product life cycle ontology for additive manufacturing. *Computers in Industry* 105 (2019), 191–203.
- [2] ARP, R., SMITH, B., AND SPEAR, A. D. *Building ontologies with basic formal ontology*. Mit Press, 2015.
- [3] BAADER, F., CALVANESE, D., MCGUINNESS, D., PATEL-SCHNEIDER, P., AND NARDI, D. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.
- [4] BELKADI, F., VIDAL, L. M., BERNARD, A., PEI, E., AND SANFILIPPO, E. M. Towards an unified additive manufacturing product-process model for digital chain management purpose. *Procedia CIRP* 70 (2018), 428–433.

- [5] BONNARD, R., MOGNOL, P., AND HASCOËT, J.-Y. A new digital chain for additive manufacturing processes. *Virtual and Physical Prototyping* 5, 2 (2010), 75–88.
- [6] BORGO, S. An ontological approach for reliable data integration in the industrial domain. *Computers in Industry* 65, 9 (2014), 1242–1252.
- [7] BORGO, S., CESTA, A., ORLANDINI, A., AND UMBRICO, A. Knowledge-based adaptive agents for manufacturing domains. *Engineering with Computers* (2018), 1–25.
- [8] BORGO, S., AND LEITÃO, P. Foundations for a core ontology of manufacturing. In *Ontologies*. Springer, 2007, pp. 751–775.
- [9] BORGO, S., AND MASOLO, C. Foundational choices in DOLCE. In *Handbook on Ontologies*, S.Staab and R.Studer, Eds. Springer Verlag Berlin Heidelberg, 2009.
- [10] BORGO, S., SANFILIPPO, E. M., SOJIC, A., AND TERKAJ, W. Ontological analysis and engineering standards: An initial study of IFC. In *Ontology Modeling in Physical Asset Integrity Management*. Springer Verlag Berlin Heidelberg, 2015.
- [11] CEUSTERS, W., AND SMITH, B. Aboutness: Towards foundations for the information artifact ontology.
- [12] CRIALES, L. E., ARISOY, Y. M., LANE, B., MOYLAN, S., DONMEZ, A., AND ÖZEL, T. Laser powder bed fusion of nickel alloy 625: experimental investigations of effects of process parameters on melt pool size and shape with spatter analysis. *International Journal of Machine Tools and Manufacture* 121 (2017), 22–36.
- [13] DINAR, M., AND ROSEN, D. W. A design for additive manufacturing ontology. *Journal of Computing and Information Science in Engineering* 17, 2 (2017), 021013.
- [14] EDDY, D., KRISHNAMURTY, S., GROSSE, I., PERHAM, M., WILEDEN, J., AND AMERI, F. Knowledge management with an intelligent tool for additive manufacturing. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2015), American Society of Mechanical Engineers.
- [15] EL KADIRI, S., AND KIRITSIS, D. Ontologies in the context of product lifecycle management: state of the art literature review. *International Journal of Production Research* 53, 18 (2015), 5657–5668.
- [16] EYERS, D. R., AND POTTER, A. T. Industrial additive manufacturing: A manufacturing systems perspective. *Computers in Industry* 92 (2017), 208–218.
- [17] FERNANDES, R., GROSSE, I., KRISHNAMURTY, S., AND WILEDEN, J. Design and innovative methodologies in a semantic framework. In *ASME 2007 International Design*

Engineering Technical Conferences and Computers and Information in Engineering Conference (2007), American Society of Mechanical Engineers, pp. 237–247.

- [18] GIBSON, I., ROSEN, D., AND STUCKER, B. *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer, 2014.
- [19] GROUP, W. O. W. OWL 2 web ontology language document overview.
- [20] GRÜNINGER, M. Using the PSL ontology. In *Handbook on Ontologies*. Springer, 2009, pp. 423–443.
- [21] GU, D., MEINERS, W., WISSENBAACH, K., AND POPRAWA, R. Laser additive manufacturing of metallic components: materials, processes and mechanisms. *International materials reviews* 57, 3 (2012), 133–164.
- [22] GUARINO, N., OBERLE, D., AND STAAB, S. What is an ontology? In *Handbook on ontologies*. Springer, 2009, pp. 1–17.
- [23] GUARINO, N., AND WELTY, C. A. An overview of OntoClean. In *Handbook on ontologies*. Springer, 2004, pp. 151–171.
- [24] GUIZZARDI, G. *Ontological foundations for structural conceptual models*. Telematica Instituut Fundamental Research Series, 2005.
- [25] HAGEDORN, T. J., KRISHNAMURTY, S., AND GROSSE, I. R. A knowledge-based method for innovative design for additive manufacturing supported by modular ontologies. *Journal of Computing and Information Science in Engineering* 18, 2 (2018), 021009.
- [26] HERRE, H. General formal ontology (GFO): A foundational ontology for conceptual modelling. In *Theory and applications of ontology: computer applications*. Springer, 2010, pp. 297–345.
- [27] HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B., DEAN, M., ET AL. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission* 21 (2004), 79.
- [28] HUANG, Y., LEU, M. C., MAZUMDER, J., AND DONMEZ, A. Additive manufacturing: current state, future potential, gaps and needs, and recommendations. *Journal of Manufacturing Science and Engineering* 137, 1 (2015), 014001.
- [29] IMRAN, M., AND YOUNG, B. The application of common logic based formal ontologies to assembly knowledge sharing. *Journal of intelligent manufacturing* 26, 1 (2015), 139–158.

- [30] KEET, C. M. The use of foundational ontologies in ontology development: an empirical assessment. In *Extended Semantic Web Conference* (2011), Springer, pp. 321–335.
- [31] KIM, D. B., WITHERELL, P., LIPMAN, R., AND FENG, S. C. Streamlining the additive manufacturing digital spectrum: A systems approach. *Additive manufacturing* 5 (2015), 20–30.
- [32] LIANG, J. S. An ontology-oriented knowledge methodology for process planning in additive layer manufacturing. *Robotics and Computer-Integrated Manufacturing* 53 (2018), 28–44.
- [33] MASOLO, C., AND BORGIO, S. Qualities in formal ontology. In *Foundational Aspects of Ontologies (FOnt 2005) Workshop at KI* (2005), pp. 2–16.
- [34] MASOLO, C., BORGIO, S., GANGEMI, A., GUARINO, N., AND OLTRAMARI, A. Wonderweb deliverable D18. Tech. rep., Laboratory for Applied Ontology ISTC-CNR, 2003.
- [35] MIES, D., MARSDEN, W., AND WARDE, S. Overview of additive manufacturing informatics: “a digital thread”. *Integrating Materials and Manufacturing Innovation* 5, 1 (2016), 6.
- [36] MIZOGUCHI, R. Part 3: Advanced course of ontological engineering. *New Generation Computing* 22, 2 (2004), 193–220.
- [37] NEGRI, E., FUMAGALLI, L., GARETTI, M., AND TANCA, L. Requirements and languages for the semantic representation of manufacturing systems. *Computers in Industry* 81 (2016), 55–66.
- [38] PALMER, C., USMAN, Z., CANGIGLIERI JUNIOR, O., MALUCELLI, A., AND YOUNG, R. I. Interoperable manufacturing knowledge systems. *International Journal of Production Research* 56, 8 (2018), 2733–2752.
- [39] PRUDHOMMEAUX, E. Sparql query language for RDF. <http://www.w3.org/TR/rdf-sparql-query/> (2008).
- [40] QI, Q., PAGANI, L., SCOTT, P. J., AND JIANG, X. A categorical framework for formalising knowledge in additive manufacturing. *Procedia CIRP* 75 (2018), 87–91.
- [41] RIJGERSBERG, H., VAN ASSEM, M., AND TOP, J. Ontology of units of measure and related concepts. *Semantic Web* 4, 1 (2013), 3–13.
- [42] ROH, B.-M., KUMARA, S. R., SIMPSON, T. W., MICHALERIS, P., WITHERELL, P., AND ASSOUROKO, I. Ontology-based laser and thermal metamodels for metal-based

- additive manufacturing. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2016), American Society of Mechanical Engineers.
- [43] SADEGHI, S., AND AMERI, F. An intelligent process planning system based on formal manufacturing capability models. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2013), American Society of Mechanical Engineers.
 - [44] SANFILIPPO, E. M., BENAVENT, S., BORGO, S., GUARINO, N., TROQUARD, N., ROMERO, F., ROSADO, P., SOLANO, L., BELKADI, F., AND BERNARD, A. Modeling manufacturing resources: An ontological approach. In *IFIP International Conference on Product Lifecycle Management* (2018), Springer, pp. 304–313.
 - [45] SANFILIPPO, E. M., AND BORGO, S. What are features? an ontology-based review of the literature. *Computer-Aided Design 80* (2016), 9–18.
 - [46] SANFILIPPO, E. M., JEANSON, L., BELKADI, F., LAROCHE, F., AND BERNARD, A. A foundational view on nominal and actual qualities in engineering. In *Formal Ontology in Information Systems: Proceedings of the 10th International Conference (FOIS)* (2018), vol. 306, IOS Press.
 - [47] SANFILIPPO, E. M., MASOLO, C., AND PORELLO, D. Design knowledge representation: An ontological perspective. In *Proceedings of AIDE. CEUR workshops proceedings* (2015), vol. 1473.
 - [48] SCHULZ, S., MARTÍNEZ-COSTA, C., KARLSSON, D., CORNET, R., BROCHHAUSEN, M., AND RECTOR, A. L. An ontological analysis of reference in health record statements. In *FOIS* (2014), pp. 289–302.
 - [49] SHI, Y., ZHANG, Y., BAEK, S., DE BACKER, W., AND HARIK, R. Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Computer-Aided Design and Applications* (2018), 1–12.
 - [50] SOLANO, L., ROSADO, P., AND ROMERO, F. Knowledge representation for product and processes development planning in collaborative environments. *International Journal of Computer Integrated Manufacturing 27*, 8 (2014), 787–801.
 - [51] TERKAJ, W., AND URGO, M. Ontology-based modeling of production systems for design and performance evaluation. In *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on* (2014), IEEE, pp. 748–753.
 - [52] THOMPSON, M. K., MORONI, G., VANEKER, T., FADEL, G., CAMPBELL, R. I., GIBSON, I., BERNARD, A., SCHULZ, J., GRAF, P., AHUJA, B., ET AL. Design for

additive manufacturing: Trends, opportunities, considerations, and constraints. *CIRP annals* 65, 2 (2016), 737–760.

- [53] WANG, X., GUARINO, N., GUIZZARDI, G., AND MYLOPOULOS, J. Towards an ontology of software: A requirements engineering perspective. In *Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS)* (2014), P. Garbacz and O. Kutz, Eds., IOS Press, Amsterdam.
- [54] WANG, Y., BLACHE, R., AND XU, X. Selection of additive manufacturing processes. *Rapid Prototyping Journal* 23, 2 (2017), 434–447.
- [55] WITHERELL, P., FENG, S., SIMPSON, T. W., SAINT JOHN, D. B., MICHALERIS, P., LIU, Z.-K., CHEN, L.-Q., AND MARTUKANITZ, R. Toward metamodels for composable and reusable additive manufacturing process models. *Journal of Manufacturing Science and Engineering* 136, 6 (2014), 061025.
- [56] WITHERELL, P., LU, Y., AND JONES, A. Additive manufacturing: A transdisciplinary experience. In *Transdisciplinary Perspectives on Complex Systems*. Springer, 2017, pp. 145–175.
- [57] YOUNG, R., GUNENDRAN, A. G., CUTTING-DECELLE, A.-F., AND GRUNINGER, M. Manufacturing knowledge sharing in PLM: A progression towards the use of heavy weight ontologies. *International Journal of Production Research* 45, 7 (2007), 1505–1519.
- [58] ZHANG, Y., AND BERNARD, A. Am feature and knowledge based process planning for additive manufacturing in multiple parts production context. In *Proceedings of 25th Annual International Solid Freeform Fabrication Symposium* (2014), pp. 1259–1276.
- [59] ZHANG, Y., AND BERNARD, A. An integrated decision-making model for multi-attributes decision-making (madm) problems in additive manufacturing process planning. *Rapid Prototyping Journal* 20, 5 (2014), 377–389.

Appendix

The knowledge base used in the examples contains various individuals, among which two cases, *cs_1* and *cs_2*, such that *cs_1* specifies information about a product made of nickel alloy 625 by a FormUp machine, see (f6)–(f10), whereas *cs_2* specifies information about a product made of titanium alloy Ti6A4V by a EOSINT M290 machine, see (f12)–(f16). Also, by (f11), the process template *ptl_1* specifies a maximum scan speed value of 750 mm/s, whereas the scan speed value established by *ptl_2* is of 875 mm/s by (f17).

- f6** CASE(*cs_1*)
- f7** HASPRODUCTTEMPLATE(*cs_1*, *ptl_1*)
- f8** HASPROCESSTEMPLATE(*cs_1*, *pstl_1*)
- f9** SPECIFIESMATERIALTYPE(*ptl_1*, *nickelAlloy625Type*)
- f10** SPECIFIESMACHINE TYPE(*pstl_1*, *formUpType*)
- f11** SPECIFIESMAXSCAN SPEEDINMILLIMETREPERSECOND(*pstl_1*, 750)
- f12** CASE(*cs_2*)
- f13** HASPRODUCTTEMPLATE(*cs_2*, *ptl_2*)
- f14** HASPROCESSTEMPLATE(*cs_2*, *pstl_2*)
- f15** SPECIFIESMATERIALTYPE(*ptl_2*, *titaniumAlloyTi6A4VType*)
- f16** SPECIFIESMACHINE TYPE(*pstl_2*, *eosintm290Type*)
- f17** SPECIFIESMAXSCAN SPEEDINMILLIMETREPERSECOND(*pstl_2*, 875)

A1. Examples of queries

The queries are expressed in SPARQL [39]. For the sake of the examples, PREFIX statements to specify the namespaces of the ontology and other relevant documents, e.g., the RDF standard specification, are omitted.

Q2 Retrieve process templates that specify a maximum speed value comprised within 800 and 900 mm/s. Result: only *pstl_2*.

```
SELECT ?processTemplate ?speedValue WHERE
  {?processTemplate a ProcessTemplate;
   specifiesMaxScanSpeedInMillimetrePerSecond ?speedValue.
  FILTER (?speedValue >= 800 && ?speedValue <= 900) }
```

Q3 Retrieve cases which specify information about products that are made of either nickel alloy type 625 or titanium alloy type Ti6AV and that have been produced by machines of type FormUp. Result: only case *cs_1*.

```

SELECT ?case WHERE
{ ?case a Case;
  hasProductTemplate ?productTemplate;
  hasProcessTemplate ?processTemplate.
  {?productTemplate specifiesMaterialType nickelAlloy625Type.}
  UNION
  {?productTemplate specifiesMaterialType titaniumAlloyTi6A4VType.}
  ?processTemplate specifiesMachineType formUpType. }

```

Q4 For each machine type in the KB, retrieve the maximum laser power value.

```

SELECT ?machineType ?powerValue WHERE
{?machineType a MachineType;
  hasMaxLaserPowerInWatt ?power.}
ORDER BY (?powerValue)

```

machineType	maxPowerValue
proXDMP200Type	300
eosIntM290Type	400
formUpType	500

Table 4: Result of (Q4)

A2. Examples of reasoning for constraints checks

Assume that formula (f18) substitutes (f9), so that the product template *ptl_1* specifies now information about a product made of material type stainless steel CX.

f18 SPECIFIESMATERIALTYPE(*ptl_1*, *stainlessSteelCXType*)

By reasoning over the knowledge base, (f18) generates an inconsistency in tandem with (f6)–(f8) and (f10), because they violate (f5).

By formula (f19), the process template *pstl_1* sets the maximum laser power value to 600 W. By (f10), *pstl_1* specifies the usage of a machine of type *FormUp*, which by (f20) has maximum laser power of 500 W (recall (Q4)).

f19 SPECIFIESMAXLASERPOWERINWATT(*pstl_1*, 600)

f20 HASMAXLASERPOWERINWATT(*formUpType*, 500)

The formulas trigger therefore rule (R3) and the user needs to revise the inserted data.