



ICPEs 2025

40th INTERNATIONAL CONFERENCE ON PRODUCTION ENGINEERING - SERBIA 2025

DOI: [10.46793/ICPEs25.009SG](https://doi.org/10.46793/ICPEs25.009SG)



Nis, Serbia, 18 - 19th September 2025

ONTOLOGIES IN MANUFACTURING

Lorenzo SOLANO-GARCÍA^{*1}

Orcid: 0000-0003-0535-314X

¹Institute of Design and Manufacturing, Universitat Politècnica de València, Spain

*Corresponding author: isolano@mcm.upv.es

Abstract: This paper presents an application of ontologies in the manufacturing field, specifically in collaborative process planning. To this end, it first examines the concept of ontology through its definition, purpose, and classification. After it presents the tools used to implement the developed ontologies and the upper ontologies used for their development. Finally, a brief description of these ontologies is provided: PPDR ontology (ontology for collaborative Development of Products and Processes focused on Resource Capabilities) and MIRC ontology (ontology for integrated Machining and Inspection process planning focusing on Resource Capabilities).

Keywords: Ontologies, manufacturing, collaborative process planning, PPDR ontology, MIRC ontology, resource capabilities.

1. INTRODUCTION

Process planning, as a link between design and manufacturing, is a key function to ensuring that the characteristics of manufactured products meet customer needs. Multiple factors converge in the tasks of manufacturing process planning. When these factors are combined with the diversity of possible strategies and approaches, they form a particularly complex scenario.

Taking as a starting point the above, this paper presents a contribution in the field of collaborative process planning. For this, an ontological approach is proposed. This ontology gives support and consistency to the co-planning tools used in creating process plans, especially in decision-making related to

the optimal and dynamics resource allocation based on resource capabilities.

2. ONTOLOGIES

This section shows an overview of ontologies, intended to provide a general and integrative vision of them. Since the relatively recent presence of ontologies in the field of engineering, and particularly in manufacturing engineering, means that they may not be sufficiently well known in these fields.

2.1 Definition

Some authors agree on the extreme difficulty of establishing a general definition for the concept of ontology, since there are many and diverse realities to which this term can

refer, usually linked to their different uses: the integration of data sets; the sharing of knowledge bases; enabling communication between software agents; decision-making support; semantic frameworks for enterprise architectures; the representation of a natural language vocabulary; the representation of semantics for complex software services and applications; or the provision of a conceptual framework for indexing content.

According to Gruber, an ontology is an explicit specification of a conceptualization [1]. This conceptualization can be understood as a set of objects, concepts, and other entities that are assumed to exist in some area of interest, along with the relationships they maintain among themselves. Terms with a precise, shared interpretation and meaning are established and used to refer to all of these entities and relationships.

Formalisms facilitate the sharing of certain common knowledge between humans and machines in a structured and precise manner, as they use a rigorous syntax for representing entities and relationships, ensuring unambiguous interpretation of their semantics. Logic uses these formalisms to reach certain conclusions through reasoning processes.

Among the definitions of ontology that are based on logical theories or formalisms, the one by Uschold and Gruninger stands out: 'an ontology is a formal description of the entities within a given domain, with the properties they have, the relationships in which they participate, the restrictions to which they are subject, and the patterns of behavior they exhibit' [2].

2.2 Purpose of ontologies

Ontologies aim to rationally establish the principles that organize and guide knowledge of being, as a part of reality, through its properties, principles, and causes.

Ontologies allow the relevant concepts and terms of a given domain to be identified and defined unambiguously. In this sense, they constitute the technological key to describing the semantics of information, overcoming the

problem of implicit and lost knowledge and enabling the exchange of semantic content.

For other authors, the main purpose of an ontology is to enable communication between computer systems, making it independent of the technologies of the individual systems, the information architectures and the application domain [3].

According to Guarino, the purpose of an ontology is the characterization of a conceptualization, with the aim of establishing a consensus about the knowledge described by that language [4]. In fact, the set of terms and definitions of an ontology are shared by all participants in the domain and, therefore, constitute a basis for communication about said domain.

However, an ontology must be understood as more than just an effective and efficient means of transmitting information. An ontology allows us to capture knowledge in a domain of interest, and although in its practical applications in engineering this has also involved data storage and maintenance. Ontologies are situated at a higher level than information systems, as they also allow reasoning based on the information they contain. Furthermore, this captured knowledge can be shared and reused for various purposes and applications related to its domain. Therefore, ontologies occupy an important place among modeling and representation techniques of knowledge.

Ontologies facilitate the representation of a body of knowledge in a formal and declarative manner, and constitute the foundation for building the bases of knowledge, representing the terminology specific to a domain and facilitating communication and the transmission of knowledge between heterogeneous agents in a knowledge-based system.

2.3 Types of ontologies and classification criteria

Based on the relationship between ontology and language, Fox and Gruninger proposed classifying ontologies into informal, semi-formal and formal, considering the degree of

precision with which both their syntax and semantics are established. In this context, natural language would be an example of an informal ontology, logical-mathematical languages would be formal ontologies and semi-informal ontologies would be found halfway between the two [5].

On the other hand, ontologies can be classified according to their degree of specificity [6]: upper ontologies, which define concepts applicable to most or all domains; core ontologies, which define concepts shared by several similar or related domains; domain ontologies, which contain concepts specific to a particular domain of interest; and application ontologies, which specialize the concepts of a domain ontology with application-specific variants (Figure 1).

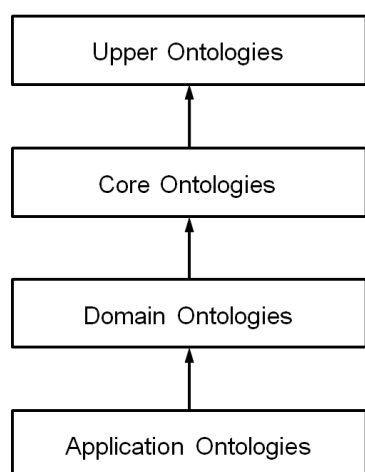


Figure 1. Types of ontologies according to the degree of specificity [6]

Taking this classification as a reference, high-level models can be obtained from the adaptation of a base ontology (upper ontology) that describes the basic concepts (foundational concepts), which are sufficiently general to be applied in a wide range of domains, and is designed so that it can be extended and meet the additional requirements introduced by a specific domain.

This aspect is fundamental when developing ontologies in engineering, because the use of different ontologies without a common basis necessarily leads to misunderstandings.

3. LANGUAGES AND TOOLS FOR THE IMPLEMENTATION OF ONTOLOGIES

This section addresses aspects related to the practical use of ontologies in engineering, from those linked to applications based on knowledge inference to those related to software tools for editing, presenting, and sharing ontologies.

3.1 Ontology editors. Protégé

There are several ontology development environments or ontology editors: OntoEdit, WebODE, Hozo, etc. Among them, Protégé stands out.

Protégé [7] offers significant features in the ontology utilization phase. Its features include: extensibility, which allows users to redefine representative primitives; a semi-automatic tool for merging and aligning ontology components; the ability to perform some tasks automatically and guide the user through the development of others; a customizable output file format capable of adapting to any formal language; a customizable user interface; and a powerful and sophisticated plug-in architecture capable of integration with other applications. Protégé, which uses, among others, the OWL and RDF syntaxes, allows the construction of domain ontologies, since, as can be seen in Figure 2, with this editor you can define classes, class hierarchies, properties, restrictions for the value of properties, relationships between classes and the properties of these relationships.

Protégé allows concepts to be described and also provides new facilities or services. It has a rich set of operators, for example, intersection, union, and negation. Since Protégé is based on a logical model that allows concepts to be defined and described, complex concepts can be built on the definitions of other simpler concepts. Furthermore, the logical model allows the use of a reasoner that can check whether all the ontology declarations and definitions are consistent with each other and can also recognize that concepts fit within each definition. The reasoner can therefore help maintain the hierarchy correctly. This is

particularly useful when dealing with cases where classes may have more than one superclass.

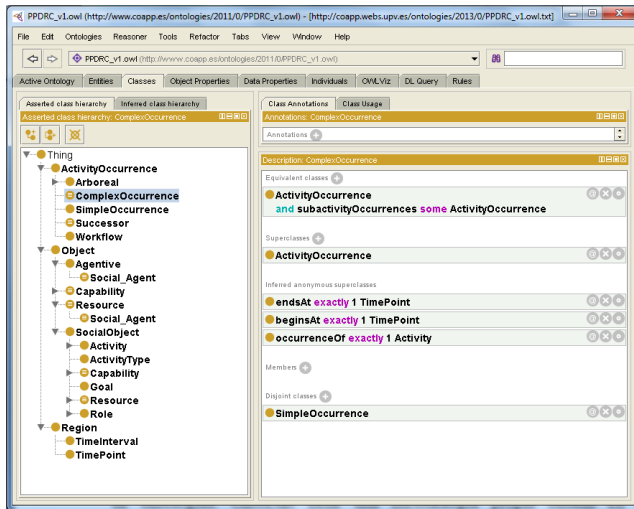


Figure 2. User interface of the Protégé ontology editor, with the PPDR ontology class taxonomy and the defining axioms of the 'ComplexOccurrence' class, which constrain its semantics

3.2 Reasoners

In addition to a suitable user interface and mechanisms to facilitate and systematize communication between collaborators, the main aid for the design and maintenance of ontology provided by current tools is an interface with reasoning systems or reasoners, most of which are based on OWL DL. Reasoners support the testing layer of the Semantic Web and can return examples of objects that meet defined constraints or perform checks to determine the veracity of an assumption. The main function of reasoners is to classify ontology concepts by establishing subclass and superclass relationships and to detect logical errors that indicate modeling flaws. That is, the validation and analysis of ontology.

Among the reasoners that work with ontologies written in OWL DL are Racer and Fact++, which are, along with Pellet, the three most commonly used reasoners with OWL. As indicated above, they check the ontology to automatically compute the classification hierarchy and also to verify its logical consistency. The class hierarchy described in

the ontology is called the *asserted hierarchy*, while the class hierarchy resulting from the reasoning process is called the *inferred hierarchy*. After the reasoner checks, it can be seen whether any class has been reclassified (i.e., if its superclass has changed) or if it is inconsistent.

4. UPPER ONTOLOGIES

Upper ontologies facilitate the integration of other, lower-level ontologies, as they allow for shared vocabulary to describe the semantics of ontological entities unambiguously and in a machine-processable manner. This is the case with DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) and PSL (Process Specification Language).

4.1 DOLCE

The DOLCE ontology is intended to support the design of domain ontologies and has been successfully used in industrial and academic projects and in diverse domains such as law, biomedicine, and agriculture. As its name suggests, 'Descriptive Ontology for Linguistic and Cognitive Engineering,' DOLCE has a clear cognitive bias and aims to capture the ontological categories that underlie human natural language and common sense. DOLCE represents the world as it is perceived by humans rather than as it is seen from the perspective of scientific theories.

DOLCE is an ontology of *particulars* in the sense that its domain of discourse is restricted to *particulars*, or entities that cannot have instances. However, *universals*, or entities that can have instances, appear in an ontology of *particulars* such as DOLCE, to the extent that they are used to organize and characterize that ontology. In other words, when designing this ontology, the classes of its taxonomy and their predicates are established (predicates are the properties of the classes and the relationships between them), which are *universals*, while the use of the ontology is based on the instances of

these classes and predicates, which are *particulars*.

DOLCE's ontological classification of the world is extensive. DOLCE's four high-level categories are: *endurant*, *perdurant*, *quality*, and *abstract* (Figure 3). *Endurants* are particulars in space, which participate in at least one *perdurant*. *Endurants* are classified as physical versus non-physical, and agentive versus non-agentive. *Perdurants* are particulars in time, which have at least one participant. *Qualities* are particulars inherent in other *endurants* or *perdurants*. *Abstracts* are entities that do not extend in space or time, have no spatial or temporal quality, and are not qualities [8].

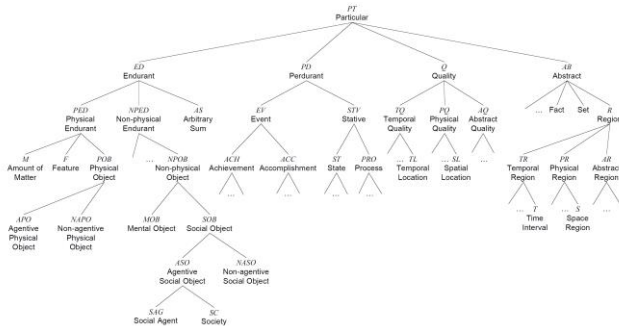


Figure 3. DOLCE Basic Categories Taxonomy [8]

4.2 PSL

PSL ontology was developed with the goal of creating a common process representation for all manufacturing applications, finding its most genuine scope in discrete manufacturing processes.

The PSL methodology is based on the identification of intuitions related to manufacturing processes, which are subsequently translated into elements of an algebraic or combinatorial mathematical structure, and finally formalized through definitions and axioms written in first-order logic that ensure rigorously developed semantics.

The PSL ontology consists of a series of interdependent modules built from a core that captures the high-level primitive concepts inherent to process specification. Each module refines this core by capturing sets of concepts

specific to a specific area related to process specification [9], [10].

PSL is the result of a project initiated in 1995 by the National Institute for Standards and Technology (NIST) to create a neutral, standardized, high-level process specification language that would allow for the integration of multiple related process applications throughout the product lifecycle, such as CAD/CAPP/CAM/CNC. This requires identifying the semantic relationships between the terms in these manufacturing applications.

Regarding its consideration as an exchange language, Figure 4 illustrates the information translation mechanism, based on the semantics of PSL and the syntax of KIF (Knowledge Interchange Format).

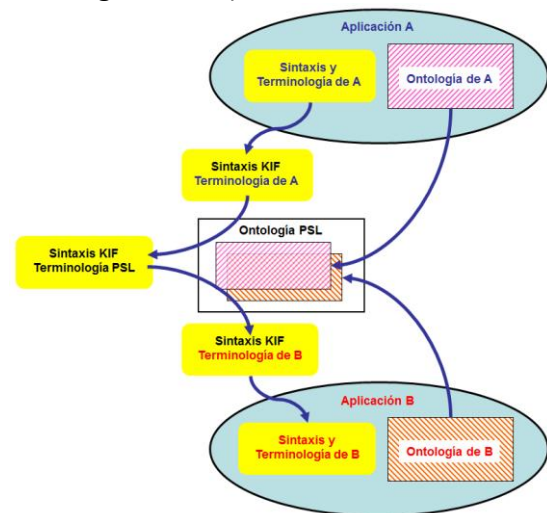


Figure 4. Using KIF syntax in the process of exchanging process information using the PSL ontology [11]

4.3 PSL-Core

The purpose of the PSL-Core is to transform into axioms a set of intuitive semantic primitives suitable for describing basic processes. These intuitive semantic primitives are: (a) there are four types of entities required for reasoning about processes: activities, activity occurrences, timepoints, and objects; (b) activities can have multiple realizations, and there may be activities that never occur; (c) points in time are ordered linearly, forward into the future and backward into the past; and (d) realizations and objects are associated with

unique points in time, which mark the start and end of the realization or object.

Despite the simplicity of PSL-Core, the set of its axioms written in the formal language of PSL provides a primitive semantics that is sufficient to describe the basic processes.

5. ONTOLOGIES IN THE MANUFACTURING DOMAIN

This section is dedicated to the PPDRC and MIRC ontologies, both focused on manufacturing process planning.

5.1 PPDRC ontology

PPDRC (Product and Processes Development Resource Capabilities) is an ontology for collaborative development of products and processes focused on resource capabilities [11].

The PPDRC ontology supports the planning of the integrated new product development process in collaborative environments deployed within the context of a virtual OKP (One-of-a-Kind Production) enterprise, where communication and cooperation between different actors is possible thanks to the high connectivity offered by Web technologies. In general terms, it can be stated that PPDRC supports the planning of any type of process, such as the manufacturing process of a product or a process. A schematic representation of the predicates of the PPDRC ontology can be seen in Figure 5.

In the case of process planning activities, the requirements outlined are particularly important, both due to their complexity, as they are subject to continuous feedback and adaptations, and due to the collaborative nature of the process and the agentic nature of the actors involved.

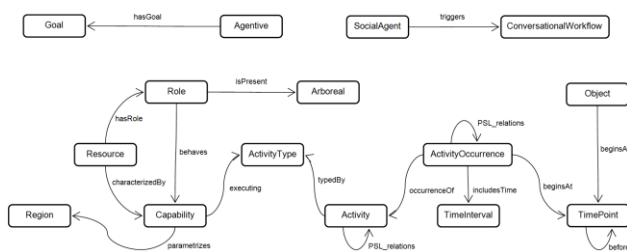


Figure 5. Predicates of the PPDRC ontology [11]

In the development of PPDRC, ontological concepts from other proposals specific to the process and resource domains are incorporated and integrated into the DOLCE+DnS Ultralite (DUL) base ontology. These include those from PSL and those from MANDATE. The use of the DUL ontology facilitates the semantic interoperability of PPDRC with ontologies from other domains.

5.2 MIRC ontology

MIRC (Manufacturing and Inspection Resource Capabilities) is an ontology for integrated machining and inspection process planning focusing on resource capabilities [11].

MIRC is a specialization of the PPDRC ontology (Figure 6). Therefore, it inherits from the latter the ability to support collaborative planning activities developed in the context of a virtual OKP and to represent the social and agentic nature of the required resources—in this case, machining and inspection equipment and tools. The specialization of PPDRC concepts provides the MIRC ontology with the semantic interoperability required for its integration with other ontologies in the manufacturing and product development domains.

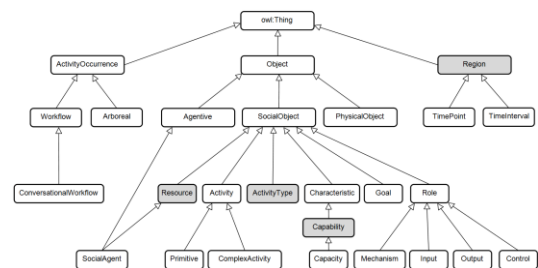


Figure 6. Basic entities of the MIRC ontology (Resource, ActivityType, Capability and Region) in the PPDRC ontology taxonomy [11]

The MIRC ontology represents both machining and inspection resources and their dimensional and geometric capabilities, as well as the activities involved in a machining and inspection plan for a part. This is because, to ensure that a part meets established quality requirements, expressed through the corresponding dimensional and geometric

requirements, it is not sufficient to assign resources compatible with the required performance level to each activity. This assignment must be made within the context of a plan, which establishes the structure (phase, subphase, and operation) and dependencies between these activities. The selection and allocation of these resources within a machining and inspection plan determines its efficiency, especially in distributed and collaborative environments, such as the integrated and collaborative product development process in a virtual OKP.

In the MIRC ontology, four entities of the PPDR ontology are specialized: Resource, ActivityType, Capability and Region (Figures 6 and 7), which represent the essential concepts used in the definition of a process plan (resources and types of activities) and those that are necessary for the allocation of resources to activities in the creation and validation of the process plan (capabilities and their quantification).

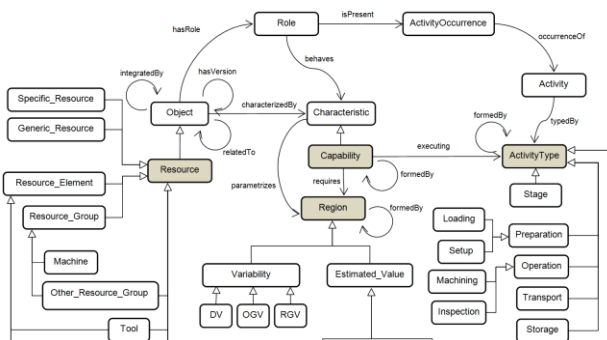


Figure 7. Predicates of the PPDR ontology, and taxonomy of the entities Resource, Region and ActivityType in the MIRC ontology [11]

In the ontology, a resource is an object that can perform an activity by exhibiting active behavior. However, it can also exhibit passive behavior when, while involved in the activity, it is not responsible for its execution. Active behavior corresponds to a mechanism-type participation role in the activity, while input, output, and control roles correspond to passive behavior.

In the first of these behaviors (active), the resource transmits the values that quantify its

capabilities to the characteristics of the resulting object. This transmission of characteristics in the execution of the activity materializes through an interface between the resource and the processed object, which introduces additional dispersion conditioned by the type of control under which the activity is executed. In the second behavior (passive), the resource is the object that receives the activity, as occurs in Preparation-type activities oriented toward resource configuration.

The MIRC ontology considers two types of activities: Operation activities and Preparation activities. The former include activities performed on the part to modify its characteristics, while the Preparation type includes activities performed on resources to modify their characteristics (Figure 8).

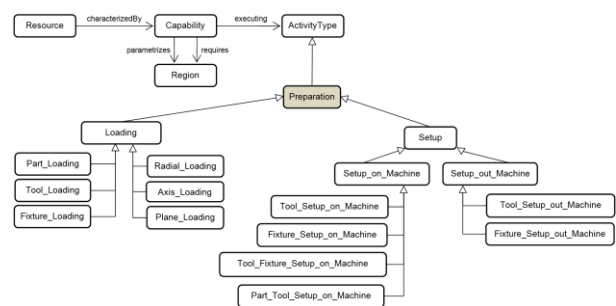


Figure 8. Preparation activities in the MIRC ontology [11]

On the other hand, resources are considered at different levels of aggregation and can be simple or complex (Figure 9). The latter are configured through Preparation activities, which modify their capacities. These capacities, for both simple and complex resources, also vary over time depending on the performance of their functions. In this sense, the preparation activities (loading and setup) of machining and inspection resources (Figure 8) are essential for determining the adequacy of the assigned resources according to technological criteria related to their dimensional and geometric capacities. Therefore, it can be stated that the MIRC ontology supports all the necessary knowledge for decision-making concerning the preparation and allocation of resources during the development of an integrated machining and inspection process plan, and is capable of

supporting the evaluation and validation of any plan.

For this evaluation and validation of the plan, a methodology is established that facilitates understanding of the conceptual framework of the MIRC ontology. This methodology is supported by graphical representations (Figure 9) that allow viewing both the plan as a whole, showing the sequence of processing activities and resource configuration, and the details of each of its component stages. Furthermore, the detailed graphical representations show the type and quantification of the characteristics associated with the resources and objects involved in the execution of the activities. Specifically, these graphs serve to show the effect of resource capabilities and activity execution (interface) on the characteristics of the resulting objects or resources.

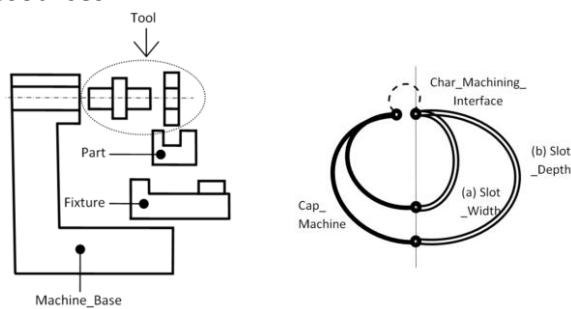


Figure 9. Physical representation of the resource participating in the machining of the slot (left) and graph with the characteristics of the machined slot (right) [11]

6. CONCLUSION

The PPDR ontology provides an efficient response to the needs arising from product development in distributed and collaborative environments.

The MIRC ontology, which is a specialization of PPDR, supports the planning of machining and inspection processes in the field of integrated and collaborative product, process, and resource development.

Both ontologies, PPDR and MIRC, have been built from upper ontologies (DOLCE and PSL), taking advantage of its semantic richness and bringing together concepts from these and

other non-ontological initiatives. Among them, the consideration of the social nature inherent to the DOLCE ontology, the treatment of resources and roles derived from the MANDATE initiative, and the description of activities and process plans derived from the PSL ontology stand out.

REFERENCES

- [1] Gruber, T. R. 1993. "A Translation Approach to Portable Ontologies." *Knowledge Acquisition* 5 (2): 199–220.
- [2] Uschold, M. y M. Gruninger. 1996. "Ontologies: Principles, Methods y Applications." *Knowledge Engineering Review* 11 (2): 93–136.
- [3] Deshayes, L. M., O. El Beqqali y A. Bouras. 2005. "The Use of Process Specification Language for Cutting Processes." *International Journal of Product Development* 2 (3): 236–253.
- [4] Guarino, N. 1997. "Understanding, Building and Using Ontologies." *International Journal of Human-Computer Studies* 46 (2–3): 293–301.
- [5] Fox, M. S. y M. Gruninger. 1998. "Enterprise modeling." *AI Magazine* 19 (3): 109–121
- [6] Bräuer, M. 2007. "Design of a Semantic Connector Model for Composition of Metamodels in the Context of Software Variability." PhD diss., Technische Universität Dresden.
- [7] Musen, M. A., R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy y S. W. Tu. 2003. "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development." *International Journal of Human-Computer Interaction* 58 (1): 89–123.
- [8] Masolo, C., S. Borgo, A. Gangemi, N. Guarino y A. Oltramari. 2003. *WonderWeb Deliverable D18. Ontology Library (final)*. Laboratory for Applied Ontology ISTC CNR.
- [9] Bock, C. y M. Gruninger. 2005. "PSL: A Semantic Domain for Flow Models." *Journal of Software and Systems Modeling* 4 (2): 209–231. doi: 10.1007/s10270-004-0066-x.
- [10] PSL. 2010. National Institute of Standards and Technology. Consultada en Diciembre.
- [11] Solano García, L. 2015. "Definition of an

integrated ontology of processes and resources
for the collaborative development of process

plans.” PhD diss., Universitat Politècnica de
València.