Semantic organization of product lifecycle information through a modular ontology

Giulia Bruno

Abstract**—**It is known that one of the main reasons of the success of manufacturing enterprises is their ability to design and maintain a coherent structure to represent their knowledge, especially the small and medium sized enterprises (SMEs), which are not often organized to manage information efficiently. Several commercial PLM systems have been developed in the last years to help companies in organizing their large amount of data, but they are rarely exploited mainly due to the high cost and difficult customization. Recent trends in literature focused on the development of semantic knowledge management systems, both to help companies in organizing and sharing their data and to allow the easy finding of information and its reuse. The aim of this paper is to develop a knowledge management system to structure the product lifecycle knowledge of SMEs based on a modular ontology. The modular ontology contains a reference ontology to represent the main concepts with their relationships, and several domain-specific ontologies to specialize and enrich the reference ontology with specific information. Particularly, this paper is focused on the specification of the reference ontology with a manufacturing process ontology module. Some examples of the usage of the system to extract information through SPARQL queries are also presented.

Keywords**—**Information retrieval, knowledge modeling, manufacturing systems, semantic model.

I. Introduction

ALTHOUGH knowledge management systems proved their applicability and benefits in many domains, the design and the maintenance of the underlying knowledge base is still a complex and costly task [1]. Especially in the manufacturing domain, the product lifecycle knowledge has to be carefully organized and stored because unreliable knowledge prevents the reuse of the results achieved utilizing human excellence [2].

However, companies (especially SMEs) usually do not use a knowledge management system, thus are not able to efficiently reuse previous information [3]. Furthermore, people who worked in an enterprise may have very valuable knowledge, and when they leave their knowledge is taken away [4]. Several commercial PLM tools were developed in the last years, but they are seen as complex software which need a huge effort to be understood and used, thus are rarely exploited by SMEs. Studies estimated that information retrieval is not efficient and around 50% of the available knowledge is not stored in information systems [5].

This work was partially supported by the EU-FP7 research project on Advanced Platform for Manufacturing Engineering and Product Lifecycle Management (amePLM, contract number 285171).

G. Bruno is with the Department of Management and Production Engineering of Politecnico di Torino, Italy (phone: 00390110907280; fax: 00390110907299; e-mail: giulia.bruno@polito.it).

In literature, a recent trend about manufacturing knowledge management is the inclusion of semantic models, both to help the company in organizing and sharing their data, and to allow the easy finding of information and its reuse. In fact, ontologies make it possible to integrate information from different abstraction levels, and they improve knowledge capture and reuse [6]. Several ontologies have already been proposed for knowledge management in manufacturing, but they mainly develop semantic models to grant interoperability among different systems or they are product-centric ontologies containing very detailed information on product features and few details about actual product management.

From the experience derived from the EU-FP7 amePLM project, it emerged that one of the main problems of SMEs is the lack of a semantic structure to link the different kinds of information generated about a product during its whole lifecycle. This information has to be easily searchable based on criteria different from company to company. Thus, the aim of this work is to design a more simple tool to support the manufacturing knowledge management, based on a modular sets of ontologies.

A single reference ontology contains the basic concepts together with the relationships among them, and a set of domain-specific ontologies are integrated to extend and specialize the basic concepts. The modular structure of ontologies allow an easy customization to include the specific knowledge that is needed time by time by a company. This paper is mainly focused on the specification of the reference ontology with the manufacturing processes knowledge.

The rest of the paper is organized as follows. Section 2 presents the recent works addressing the exploitation of ontologies in manufacturing. Section 3 and Section 4 detail the development process of the modular ontology at the basis of the proposed knowledge management system. Section 5 describes the whole architecture of the proposed knowledge management system and gives details on its implementation. Section 6 shows how the ontology is populated with concrete data and how it is queried to retrieve information. Finally, Section 7 draws conclusions and states future works.

II. Related Works

Several works focused on the development of semantic models for specific products or specific lifecycle phases in order to store information and share it among applications. However, previous works are mainly focused in representing very detailed information about specific aspects of product management for interoperability purpose. For example, [7] addressed the development of an ontological assembly model to allow the interoperability between software platforms. [8] focused on semantic integration of PLM objects based on an ontology, but they consider only concepts derived from the bill of material to create the ontology. [9] proposed an ontology to eliminate confusion of semantic concepts in the ship-building industry by using a classification trees.

Under the FP6 project PROMISE, [10] developed a semantic object model for product data and knowledge management in UML language, and then translated that model in EXPRESS language to create a new Product Lifecycle Management Standard as an extension of STEP. [11] developed a port-based ontology to represent the intended exchange of signals, energy, and materials. [12] introduced an approach to support interoperability in product data management, while [13] developed a manufacturing core ontology model to integrate design and manufacturing computer-based systems. [14] showed the benefits of applying ontologies to support knowledge sharing in PLM with a focus on assembly processes. Also approaches to develop a communication interface for different design support systems to share platform independent product, process and system related knowledge were proposed [15]. Under the FP7 project iProd, [16] developed a set of ontologies for integrating engineering data from heterogeneous IT systems.

A recent work addresses the definition of a manufacturing process ontology to formalize the semantics of manufacturing capability knowledge [17]. Also the present work has the aim of constructing a scalable and shared manufacturing process knowledge system. However, the model proposed in this paper covers not only the manufacturing processes but all the information involved in the product lifecycle. In fact, the manufacturing process is only one of the types of activities done during the product lifecycle, which starts from the design and ends with the disposal and recycling of the product. It is possible to define a knowledge base structure upon which organize the main concepts, and then specialize each one by means of ontologies at a lower level of details.

The modular organization of ontologies, starting from a core ontology which is then expanded and enriched, is a well established procedure [16][18] [19]. However, previous works are mainly focused on interoperability problems to map concepts from heterogeneous systems and they did not address the definition of a manufacturing ontology integrated in a comprehensive product lifecycle ontology.

This paper is focused on the description of the specialization of the manufacturing process ontology and how it is integrated in the reference PLM ontology. Preliminary developments of this reference ontology were presented by [20][21][22]. This work extends the reference PLM ontology by integrating a specific manufacturing ontology.

III. MODULAR ONTOLOGY

According to [14], the process to define an ontology starts from the identification of the set of relevant concepts, then proceeds with the organization of concepts in a formal model representing the ontology structure, and finally performs the implementation of the model in OWL. This procedure was followed to define the reference ontology and then again to define each domain-specific ontology.

A. Reference ontology

As described in more details in [22], the set of high level concepts needed to represent the product lifecycle management knowledge of a company are the following.

- **Production item**: either a product or a product component. Each product can be made of several components, and the same component can be used by different products. Each component can be in turn be composed by other components.
- **Characteristic**: a material, a functional characteristic or a physical characteristic (e.g., height, length, width, weight, etc) which refers to a production item.
- **Customer**: the reference person or company who ordered a product.
- **Activity**: an action executed during the lifecycle of a product by one or more resources, which has one or more files in input and produces one or more files in output.
- **File**: an electronic source of information, which is involved in an activity and refers to one or more resources or production items.
- **Resource**: an entity that is involved in the execution of an activity. It can be of two kinds, Person or Machine.
- **Role**: the role of a person denoting his/her skills in the company.

These concepts were organized in a formal model, shown in Fig.1, according to the UML class diagram formalism [23]. The Production item class is linked to the Customer class to store the customer who ordered each product. A production item is also associated to the Characteristic class to store the characteristic of the item. A production item can be a Product or a Component. A product can be made of several components. Each component can be in turn be composed by several other components.

A Product is associated with the activities of its lifecycle. There are two kinds of associations between Activity and File, due to the different kind of usage of the file done by the activity. An activity has one or more files in input and can produce one or more file as output.

To keep trace of the resources involved in the activities, the Resource class was linked to the Activity class. Also the other specialization of the Resource class, i.e., the Machine class, is linked to the Activity class to store the machines used in each activity. For each Person, it is know the Role he/she has in the company, and also the roles that can execute each activity are stored.

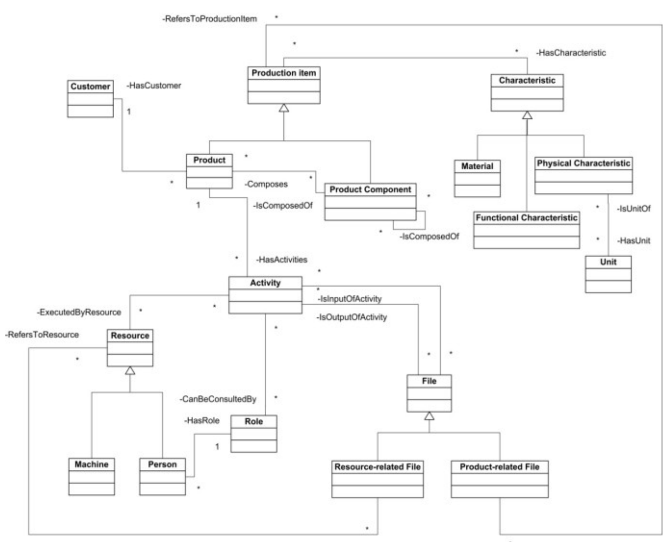

Fig.1. UML class diagram of the reference ontology.

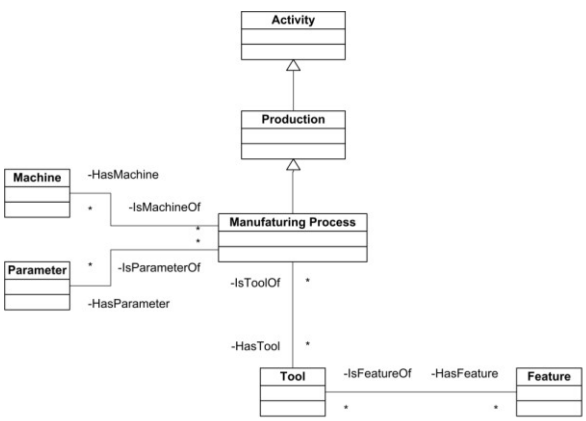

Fig.2. UML class diagram of the manufacturing process ontology.

B. Manufacturing process ontology

The reference ontology can be enriched by defining other domain-specific ontologies and integrating them by adding the proper relationships with the reference ontology. Several domain-specific ontologies were already defined to specialize different concepts of the reference ontology, i.e., the Characteristics, Material and the File (see [22]). This section is focused on the manufacturing process ontology, which specialises the Activity class. The high level model of the manufacturing process ontology is shown in Fig.2. As shown in the figure, the Manufacturing process class is a specialization of the Production class, which in turn is a specialization of the Activity class in the reference ontology.

The main class of this ontology are the following:

- **Manufacturing process**: a process executed during the production of a product.
- **Machine**: a machinery involved in a manufacturing process.
- **Parameter**: a parameter required by a manufacturing
- process.
- **Tool**: a tool used in a manufacturing process.
- **Feature**: a feature of a tool.

Then, each class is further specialized in subclasses accordingly with the terms used by [24], as described in the following.

IV. Manufacturing Process Ontology

A. Manufacturing process hierarchy

Manufacturing processes are divided into two main types: Processing operations and Assembly operations. Processing operations are processes which transform a workpiece into a more advanced state, e.g., by changing its geometry, enhancing its surface or changing its appearance, while assembly operations joint two or more pieces into a new entity. Process operations are divided in Shaping process, Property enhancing process, and Surface processing operations.

Shaping process change the geometry of a starting workpiece. Inside this category there are Solidification processes, Deformation processes, Particulate processing and Material removal. Solidification processes, which are processes that use molten material to give the workpiece a particular geometry, gather principally Casting and Molding processes. Both classes have other subclasses for the different casting and molding operations that can be executed. The Casting processes are divided in Expendable mold casting (i.e., Sand casting, Shell molding, Vacuum molding) and Permanent mold casting (i.e., Die casting, Squeeze casting and Centrifugal casting). The Molding processes are divided in Injection molding, Compression molding, Transfer molding, Blow molding, Rotational molding, and Thermoforming.

The Particulate processing operations include the procedures that create new entities from the joint of small particles, i.e., Pressing and Sintering processes.

The Deformation processes change the shape of the initial workpiece without removing material from it. The subclasses are Rolling, Forging and Extrusion. Each of them has its hierarchy of sub-processes, i.e., Shape rolling, Rolling mills, Thread rolling, Ring rolling, Gear rolling and Roll piercing for the Rolling processes, Open-die forging, Impression-die forging, Fleshless forging, Upset forging, Radial forging, Roll forging, Orbital forging, Hubbing, Isothermal Forging and Trimming for the Forming processes and Direct extrusion and Indirect extrusion for Extrusion processes.

The Material removal processes change the shape of the workpiece by removing material from it. The processes that belong to this group are Machining processes, Non traditional machining processes and Abrasive processes. Among the Machining processes there are Turning, Drilling, Milling, Shaping, Planning, Broaching, Sawing. Among the Abrasive processes there are Grinding, Honing, Lapping, Polishing, Buffing, and among Non traditional machining processing there are Mechanical Energy processes (Ultrasonic machining, Water jet cutting, Abrasive jet cutting, Abrasive jet machining, Abrasive flow machining), Electrochemical machining process (Electrochemical machining, Electrochemical grinding), Thermal energy processes (Electric discharge machine, Electric discharge wire cutting), Electron beam machining, Laser beam machining, Arc cutting processes, Oxyfuel-cutting processes and Chemical machining (Chemical milling, Chemical blanking, Chemical engraving, Photochemical machining).

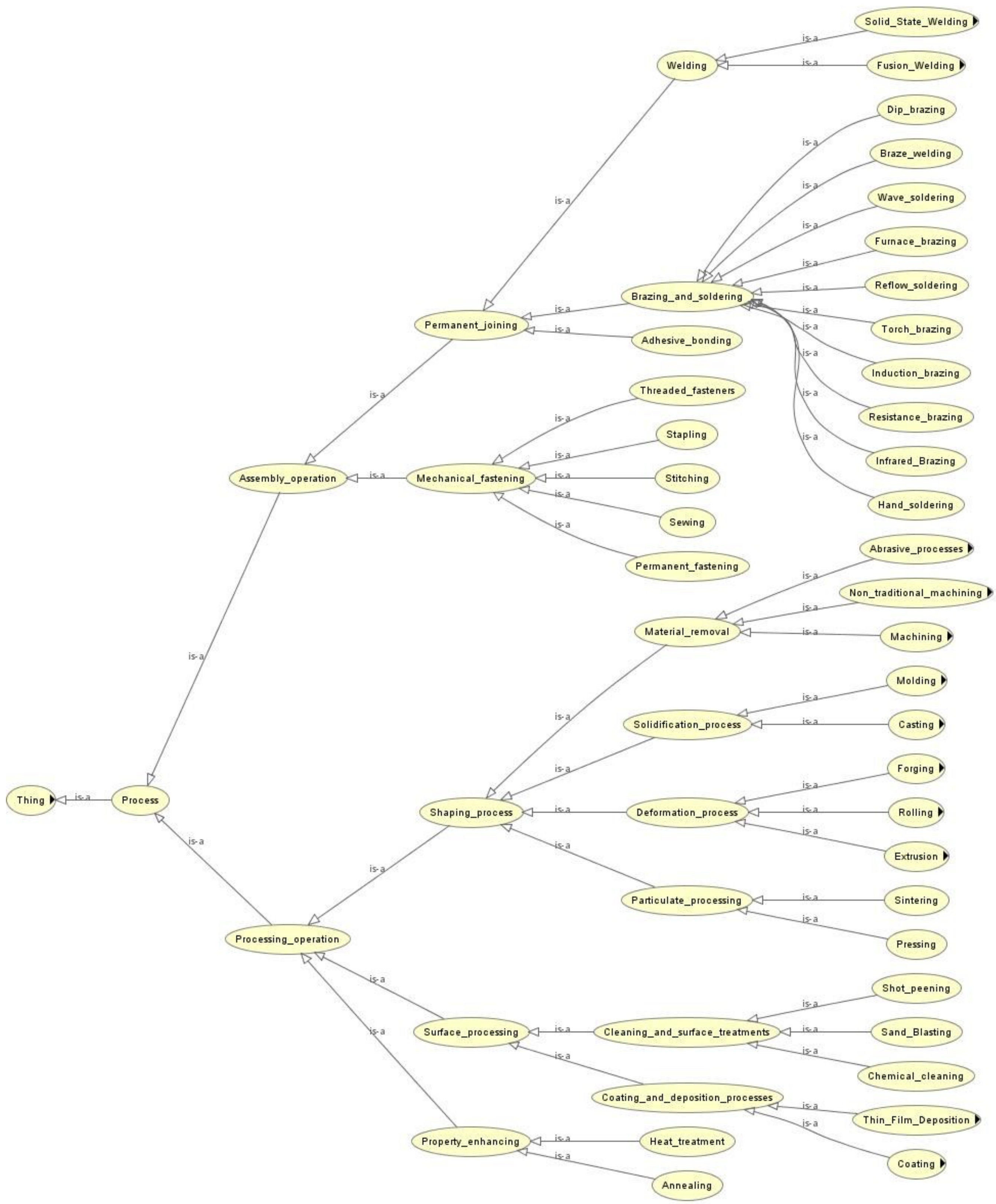

Fig.3. First four levels of the Manufacturing process hierarchy.

Property-enhancing processes improve some properties of the workpiece such as its hardness or resistance. These processes are Heat treatment and Annealing.

Finally, Surface processing is an operation that works on the surface of the workpiece to cleaning it or to add a coat to it. Processes belonging to this category are Cleaning and surface treatments (Shot peening, Sand Blasting), Coating and deposition processes (Coating, Electroplating, Anodizing of aluminum, Organic coating, Porcelain enameling) and Thin Film Deposition (Physical vapor deposition, Chemical vapor deposition).

Assembly operations are divided in Permanent joining processes and Mechanical fastening. Permanent joining processes are Welding, Brazing and soldering, and Adhesive bonding. Welding processes are further divided between Fusion welding and Solid state welding. To Fusion welding belong Arc Welding (both with Consumable electrodes such as Shielded metal arc welding, Gas Metal arc welding, Flux-Cored arc welding, Electrogas welding, Submerged arc welding, and with Non consumable electrodes such as Gas tungsten arc welding, Plasma arc welding, Carbon arc welding, and Stud welding), Resistance Welding (Resistance spot welding, Resistance seam welding, Resistance projection welding, Flash welding, Upset welding, Percussion welding, High-frequency resistance welding, Induction welding), Oxyfuel gas welding, Electron-beam welding, Laser-beam welding, Electroslag welding and Thermit welding. To Solid state welding belong Forge welding, Cold welding, Roll welding, Hot pressure welding, Diffusion welding, Explosion welding, Friction welding, Friction Stir welding, and Ultrasonic welding.

Brazing and soldering are divided in Torch brazing, Furnace brazing, Induction brazing, Resistance brazing, Dip brazing, Infrared Brazing, Braze welding, Hand soldering, Wave soldering and Reflow soldering.

The Mechanical fastening processes are Threaded fasteners, Permanent fastening methods, Stitching, Stapling and Sewing. The graphical visualization of the first four levels of the Manufacturing process hierarchy is reported in Fig. 3.

B. Machine hierarchy

This class contains the information about the equipment that each process needs. The hierarchy of concepts of this class reflects the hierarchy of the manufacturing processes because the division of the machines is based on the process in which they are used.

Thus, the Machine class is divided in Welding machine, Cleaning machine, Coating machine, Deformation machine, Machining equipment, Molding machine, Brazing and soldering machine, Heat treatment machine, Casting machine and Non traditional machine.

Welding machines are used in welding processes and are further divided in Flash welding machines, Ultrasonic machine, Plasma welding machines, Spot welding machines, Rotational stir machines, Power supplies, Controlled explosion machine, Vacuum chambers, Submerged arc welding machines, Resistance seam welding machines, Diffusion welding chambers, and Common welding machines.

Cleaning machines are divided in Abrasive blast machines and Air blast machines.

Coating machine contains all the machinery used to perform coating processes, and are divided in Tanks and Reactions chambers.

Deformation machine includes Presses, Ring rolling machines, Rolling mills and Thread rolling machines.

Machining equipment gathers the traditional machines to perform machining processes, such as Shaper, Sawing machine, Broaching machine, Planner, Drill press, Grinding machine, Lathe and Milling machine.

Molding machine is used all over the solidification processes related with molding and includes Injection molding machines, Multicavity indexing machines, Extrusion machines, and Blow molding stations.

Brazing and soldering machine, they are divided in Wave soldering machines and Induction brazing machines.

Casting machines are used in the solidification processes and inside this category there are Casting furnaces, Rotation machines and Shell molding machines.

Non-traditional machining equipment includes Water jet machines, Ultrasonic machines, CNC plasma machine, Abrasive flow machine, o Etching tank and Abrasive jet machines.

C. Tool hierarchy

Tools are the elements that directly perform the work on the workpiece. Each tool has its own features such as size, material, shape, etc. The hierarchy of this class is defined with the same goal of the Machine class, enhancing the relationship between Manufacturing processes and Tools.

The tools are firstly divided among Molds, Welding tools, Forging tools, Abrasive tools, Rolling tools, Non-traditional machining tools, Machining tools, and Extrusion dies.

Molds are rigid frames that are filled with molten material to create a new piece. Molds are divided between Molding molds and Casting molds. Molding molds are used with polymers and include Rotating molds, Compression molds, Injection molds, Transfer molds, Thermo-forming molds, and Blow molds. Casting molds are used with molten metal, and can be Expendable molds (if they cannot be re-utilized) or Permanent molds (if the same mold can be used many times). Expendable molds are Ceramic shell molds and Sand molds. Permanent molds instead are Metal molds, Rotational molds and Die molds.

Welding tools are used in different welding processes and include Welding molds, Solid state tools, and Fusion welding tools. Solid-state tools are Compression rolls, Rotation tools, and Ultrasonic transducers. Fusion welding tools are mostly related with Electrodes, which can be Consumables, if they are consumed during the process or Non-consumables. Consumables electrodes are Coated electrodes, Electrogas welding electrode, Flux cored electrodes, Wire electrodes, and GMAW guns. Non-consumable electrodes are mostly

Tungsten electrodes. Another tool used by Fusion welding processes it the Resistance-welding tool.

Fig.4. First four levels of the Tool hierarchy.

In Forging process, the Forging tools used are Forging hammers, Press rams, Forging roll, Forging dies and Rotation dies.

Abrasive tools include Laps, Grinding tools, and Horning rolls.

In rolling processes, the Rolling tools used are Shaping rolls, Mandrels, Common rolls and Shaped dies.

Then there are the tools used in machining, both traditional and non traditional. The Traditional machining tools include Broaching tools, Drilling bits, Cutting tools, Saw blades, Turning tools and Milling cutters.

The Non traditional machining tools are Hand held nozzles, Oxyfuel torches, Plasma arc torches, Laser guns, Photochemical tools, Nozzles, Carbon electrodes and Electron beam gun.

The graphical visualization of the first four levels of the tools hierarchy is reported in Fig. 4.

D. Parameter and Feature hierarchy.

The Parameter class represents the parameters of the manufacturing processes that have to be stored, such as the speed of the drill, the temperature, the water pressure, etc. The taxonomy is organized in five kinds of parameters: Pressure, Sand mixture, Speed, Time and Temperature.

The Feature class represents the features of the tool used in each manufacturing process. These features are Dimension and Material.

E. Relationships

The main relationships of the manufacturing process model are the ones represented in Fig.2, i.e., hasMachine, hasParameter, hasTool and hasFeature. These relationships are implemented as properties in the ontology. Each property can have its own hierarchy of sub-property, as each class has its hierarchy of subclasses.

The property hasTool links the Process class with the Tool class, and also all their subclasses. The hierarchy of this property follows the Tool class and each manufacturing process is linked with a tool with a property called has<NameOfTheTool>. For example, the first division of the Tool class is in Molds, Welding tools, Forging tools, Abrasive tools, Rolling tools, Non traditional machining tools, Machining tools, and Extrusion dies. Thus, the first division of the hasTool property is hasMoldTool, hasWeldingTool, hasForgingTool, hasAbrasiveTool, hasRollingTool, hasNontraditionalMachiningTool, hasMachiningTool and hasExtrusionDie. The same procedure was applied to the other levels of the hierarchy.

The property hasMachine links the Manufacturing process class with the Machine class. Similar to the case of hasTool property, the hierarchy of the hasMachine property is analogous to the Machine class, the only difference is that each subproperty has a name has<NameOfMachine>. Thus, the hasMachine class is divided in hasBrazingAndASolderingMachine, hasCastingMachine, hasCleaningMachine, hasCoatingMachine, hasDeformationProcessMachine, hasHeatTreatmentMachine, hasMachiningMachine, hasMoldingMachine, hasNonTraditionalMachiningMachine, hasWeldingMachine.

The hasParameter property links a Manufacturing process with one or more Parameters. The hierarchy of this property contains the subproperties of hasPressureParameter, hasSandMixtureParameter, hasSpeedParameter, hasTimeParameter, and hasTemperatureParameter. The hasPressureParameter links a process with a parameter of pressure. The hasSandMixtureParameter property is used to link a process that use sand molds to the parameter of sand mixture, which indicates the composition of the sand used to make the mold. The hasSpeedParameter is a property that assigns a speed kind parameter to a process. The hasTimeParameter links a process with a time parameter, and the hasTemperatureParameter make the connection between the Temperature class with certain processes that uses this parameter.

The hasFeature relationship links a Tool with its Features. It has only two subproperties, hasDimension, which assigns a Dimension feature to a Tool, and hasMaterial which assigns a Material feature to a tool.

V. Implementation of the Knowledge Management System

The conceptual architecture of the proposed knowledge management system based on the modular ontology is shown in Fig.5. As described in the previous section, the core element is the modular ontology, which includes the reference ontology and several domain-specific ontologies. Each ontology is used both as a data model to describe the relationships among data and as taxonomy to specify the classes and subclasses of concepts. The users can insert data in the ontology through the Ontology population functionality or can query the ontology to retrieve data through the Ontology query functionality.

The system was developed using Jena, an open source Java framework for building semantic applications, which offers different tools including APIs for RDF and ontologies, reasoners and support for SPARQL queries [25]. Fig.6 shows the Jena methods used to built the ontology.

To create a new ontology through Jena first a new Ontology model is created, which is an object of the OntModel Class, together with the prefix that will be used to construct the URI of each resource of the model, as reported in the upper part of Fig.6 (first three lines). Once the new ontology is created, it is possible to define its classes and properties. A new class is created as an object of OntClass using the createClass method of OntModel. The figure shows the creation of the Manufacturing process class. Once the class is created, the hierarchy of its subclasses is built using the addSubClass method of OntClass, such as the Processing operation shown in the figure.

To implement a property between classes, firstly the property has to be created as an object of OntProperty, and then is necessary to define its domain and range, as shown in the lower part of Fig.6. Once all the classes and relationships have been defined, the OntModel can be printed in an OWL file containing the whole manufacturing ontology. This file can be visualized by using Protégé, an open source ontology editor tool [26]. The ontology visualized in Protégé is shown in Fig.7.

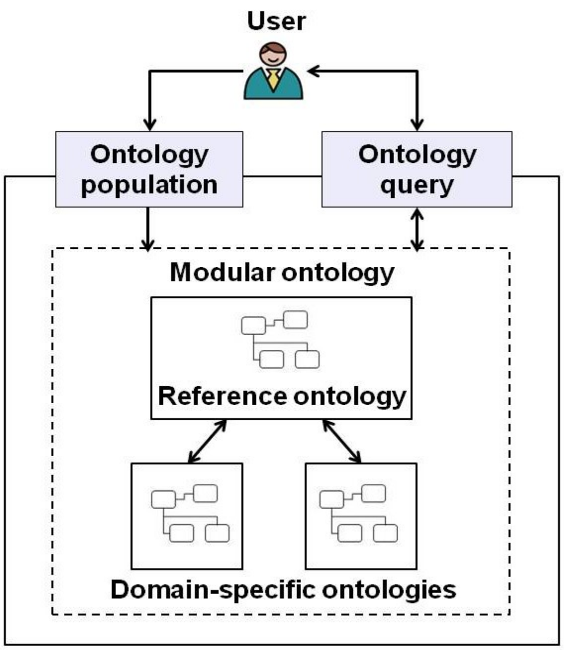

Fig.5. Conceptual architecture of the proposed knowledge management system.

```
OntModel m= ModelFactory.createOntologyModel();

String ns= "http://www.owl-ontologies.com/Manufacturing_process.owl#";
m.setNsPrefix("", ns);

    OntClass  Manufacturing_process = m.createClass(ns+"Manufacturing_process");

        OntClass Processing_operation= m.createClass(ns+"Processing_operation");
        Manufacturing_process.addSubClass(Processing_operation);


        OntClass Shaping_process =m.createClass(ns+"Shaping_process");
        Processing_operation.addSubClass(Shaping_process);

        OntClass Solidification_process =m.createClass(ns+"Solidification_process");
        Shaping_process.addSubClass(Solidification_process);

            OntClass Casting =m.createClass(ns+"Casting");
            Solidification_process.addSubClass(Casting);
            Casting.addLabel("fundicion", "Spanish");
```

```
        OntProperty hasTool = m.createTransitiveProperty(ns+"hasTool");
        hasTool.addDomain(Manufacturing_process);
        hasTool.addRange(Tool);
        Manufacturing_process.addProperty(hasTool, Tool);

            OntProperty hasMoldTool = m.createTransitiveProperty(ns+"hasMoldTool");
            hasMoldTool.addDomain(Solidification_process);
            hasMoldTool.addRange(Mold);
            Solidification_process.addProperty(hasMoldTool, Mold);
            hasTool.addSubProperty(hasMoldTool);
```

Fig.6. Portion of the Jena program used to create the manufacturing ontology.

VI. Population and Query of the Ontology

Once the ontology structure was defined, it can be populated with instances, i.e., concrete data belonging to the defined classes. As example, let consider two groups of processes. The first one is composed by three drilling processes and the second one by five water-cutting processes.

The three drilling processes are the following.

- Drilling of the piece N109: the tool used is a titanium bit with a measure of 1/2 inches, the equipment is a floor drill press made by Northern tools, and the parameter is the rotation speed, which is set to 900 rpm.
- Drilling of the piece N103: the tool used is a 5/8 inch cobalt bit, the equipment is a Benchtop drill press, the speed is set in the level 4 of the machine.
- Drilling of the piece N101: the tool used is a 9/16 inches cobalt bit, the equipment is a JET floor drill press and the speed is fixed in level 3.

The five water-cutting processes are the following.

- Water cutting 1: The tool is a nozzle of 7/15 inches, the machine is a Maxiem 2030 machine, and the water pressure is 3 MPa.
- Water cutting N2: The nozzle is 7/15 inches and also uses a Maxiem 2030 but the water pressure is 5.5 MPa.
- Water cutting N3: The nozzle of this process is 6/15 inches, the model is an Edge Xs and the water pressure is set at 4 MPa.
- Water cutting N4: The nozzle also is of 7/15 inches, the model is an Edge Xs and the pressure is 3.5 MPa.
- Water cutting N5: The nozzle is 1/3 inches and the model Edge Xs, the pressure is set in 6 Mpa.

To create an individual of a certain class firstly an object of Individual has to be created by using the createIndividual method of the Jena library. After the individuals have been created, they are linked through one of the existing properties. The code used to create the instances related to one process for each group is reported in Fig.8.

After these data were inserted as ontology instances, they can be used for information retrieval by using SPARQL queries.

A first information to retrieve can be the dimension of the hole performed for all the drilling processes available in the system. Fig. 9 reports the structure of the SPARQL query to retrieve this information and the results after its execution.

The first part of the query contains the prefixes used to locate the resources. Then the rest of the query retrieve the required information. Particularly, the SELECT command is followed by ”*” to retrieve the values of all the variables. Queries in SPARQL identify the variables by putting a question mark before their name. The conditions are represented by triples of subject, predicated and object.

The first triple inside the WHERE statement means that the query will search for all the resources linked with the ”hasDrillbit” property, i.e., instances of the class Drilling processes linked to the instances of the class Drill bit. The second triple is used to find the features linked with the drill bits through the ”hasFeature” property among the data retrieved by the previous triple. The results of this SPARQL query are the name of all the drilling processes, the tool used and the feature of each tool, as reported in the box.

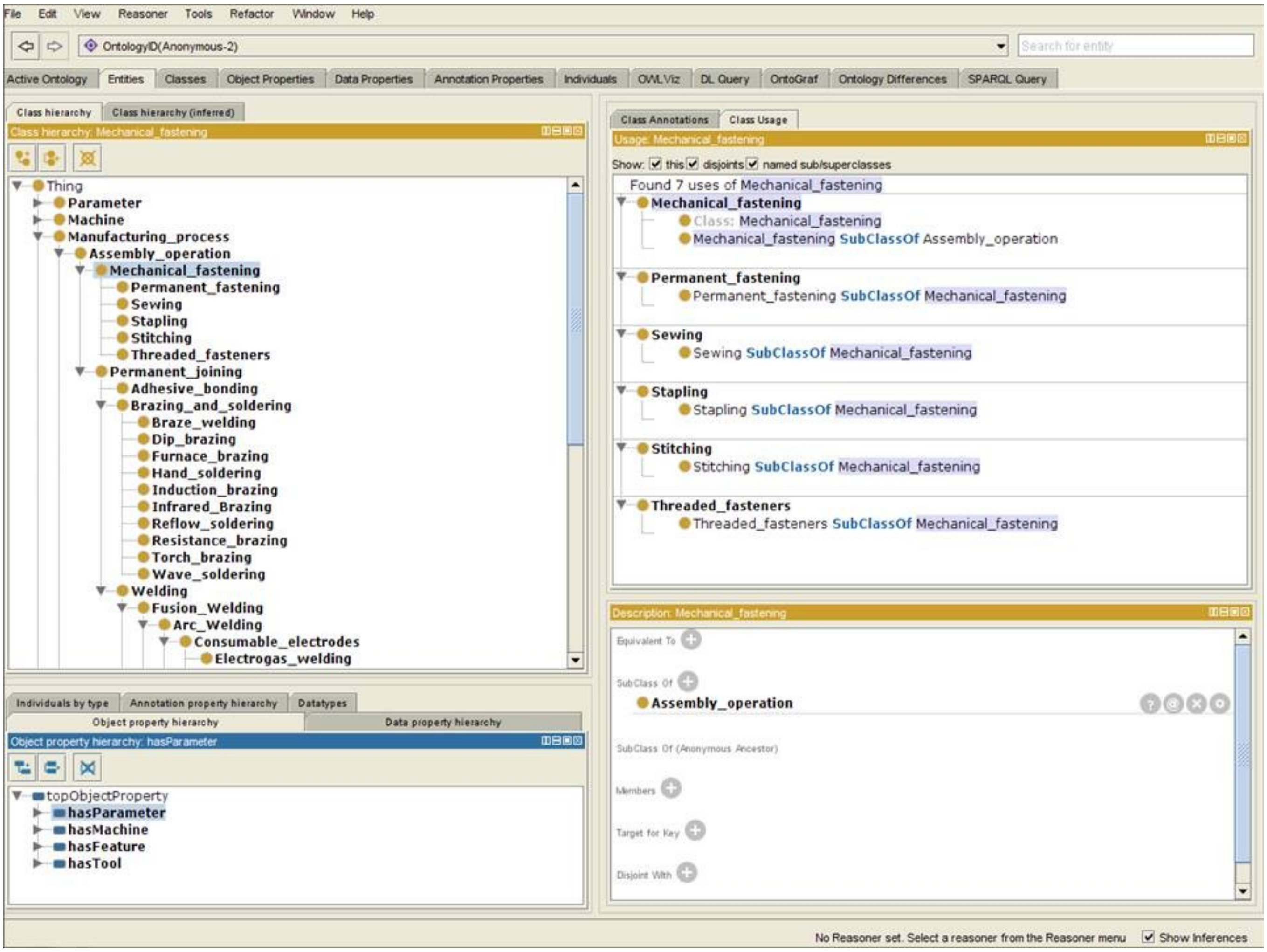

Fig.7. Screenshot of the manufacturing ontology in Protégé.

```
Individual Drilling_piece_N101 = m.createIndividual(ns+"Drilling_piece_N101",Drilling);
Individual JET_Floor_Drill_Press = m.createIndividual(ns+"JET_Floor_Drill_Press",Drill_press);
Drilling_piece_N101.addProperty(hasDrill_press,JET_Floor_Drill_Press );

Individual Cobalt_Drill_Bit = m.createIndividual(ns+"Cobalt_Drill_Bit",Drill_bit);
Drilling_piece_N101.addProperty(hasDrill_bit,Cobalt_Drill_Bit );
Individual Speed_3 = m.createIndividual(ns+"380rpm",Speed_parameter);
Individual hole_measure1 = m.createIndividual(ns+"9/16in",Feature);

Drilling_piece_N101.addProperty(hasSpeed_parameter,Speed_3 );
Cobalt_Drill_Bit.addProperty(hasFeature,hole_measure1 );
```

```
Individual Water_cutting_N1 = m.createIndividual(ns+"Water_cutting_N1",Water_jet_cutting);
Individual Nozzle01 = m.createIndividual(ns+"Nozzle01",Nozzle);
Individual MAXIEM_2030_01 = m.createIndividual(ns+"MAXIEM_2030_01",Water_jet_machine);
MAXIEM_2030_01.addComment("MAXIEM 2030 JetMachining Center", "en");
Individual Water_pressure01 = m.createIndividual(ns+"3MPa",Water_pressure);
Water_pressure01.addLabel("3000000",null);
Individual Nozzle01_dimension = m.createIndividual(ns+"7/15_01",Feature);

Water_cutting_N1.addProperty(hasNozzle,Nozzle01 );
Water_cutting_N1.addProperty(hasWater_jet_machine,MAXIEM_2030_01 );
Water_cutting_N1.addProperty(hasWater_pressure,Water_pressure01 );
Nozzle01.addProperty(hasFeature,Nozzle01_dimension );
```

Fig.8. Portion of the Jena program used to create the instances of the manufacturing ontology.

Another request can regards the water jet cutting processes. If a new workpiece arrives and it is known that to cut it the process must have a pressure of at least 4MPa, there is the need to know which of the processes that the plant is currently doing are able to perform the cut of the workpiece. Since for each water jet cutting process it is known its own water pressure parameter, it is possible to define a SPARQL query to retrieve this information, as reported in Fig.10.

In this case two variables have to be selected, the name of the water cutting process and the pressure used by the process. Also in this case there are two triples, the first one to select all the resources linked by the ”hasWaterPressure” property (i.e., the individuals from the Water cutting process class and the of the Water pressure class). Then the second triple selects the resources linked by the ”label” property. These resources are the members of the Water pressure class and their label. The FILTER command is used to filter data, to select only the processes with a pressure over 4 MPa.

At the moment the SPARQL queries were executed directly through Protégé, but ad-hoc user interface can be implemented to execute them through in more simple ways. The information retrieved can be related not only to manufacturing operations but also to manufacturing operations linked with specific product, by using the manufacturing ontology integrated with the reference ontology.

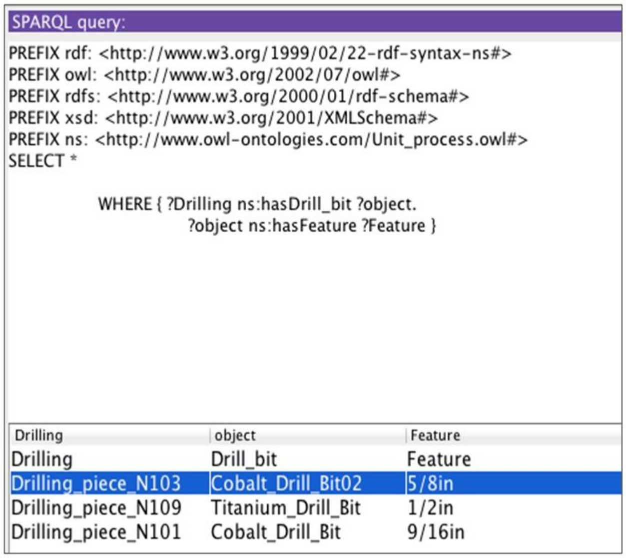

Fig.9. SPARQL query to retrieve the information of all the drilling processes stored in the ontology.

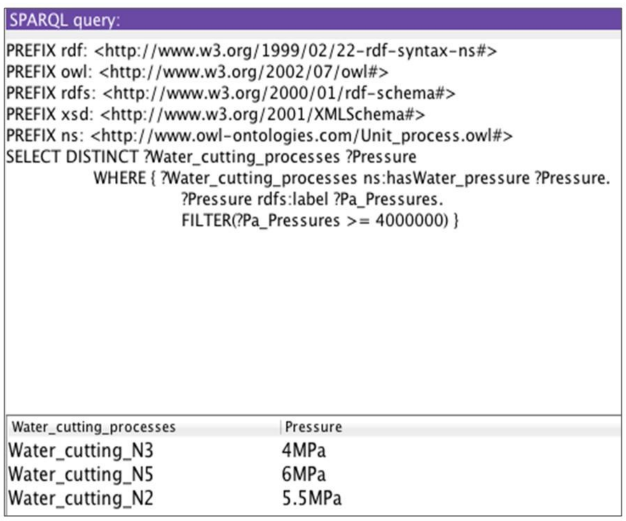

Fig.10. SPARQL query to retrieve all the water jet cutting processes using a pressure higher or equal to 4 MPa stored in the ontology.

VII. Conclusion

This paper proposes a knowledge management system to store and reuse the manufacturing knowledge available in companies. This system is based on a modular ontology, which contain a core reference ontology which can be enriched with several domain-specific ontologies. Particularly, in this paper, the integration with a manufacturing process ontology was shown. The main advantage of using this model is that it is general enough to be adopted in different application domains and its hierarchical structure allow the definition of multiple domain-specific ontologies, each of them importing the general concepts, and thus coherent with the others. This model is currently under evaluation in different application domains to show its flexibility and potentiality, and specific user interfaces are under development to allow the automatic population and query of the ontology.

References

[1] J. Baumeister, J. Reutelshoefer, and F. Puppe, “Engineering intelligent systems on the knowledge formalization continuum”, International Journal of Applied Mathematics and Computer Science 21(1), 2011, pp. 27–39.
[2] L. Horvath, and I. Rudas, “New method of knowledge representation and communication for product object modeling”, Latest Advances in Information Science and Applications, 2012, pp. 75–80.
[3] J. LeDuigou, A. Bernard, N. Perry, and J. Delplace, “Generic plm system for smes: application to an equipment manufacturer”, International Journal of Product Lifecycle Management 6(1), 2012, pp. 51–64.
[4] N. Huang, and S. Diao, “Ontology-based enterprise knowledge integration”, Robotics and Computer-Integrated Manufacturing 24(4), 2008, pp. 562–571.
[5] B. Kheder and H. Bouras, “An analysis of the interaction among design, industrialization and production”, International Conference on Product Lifecycle Management, 2010.
[6] X. Chang, A. Sahin, and J. Terpenny, “An ontology-based support for product conceptual design”, Journal of Robotics and Computer-Integrated Manufacturing 24, 2006.
[7] X. Fiorentini, I. Gambino, V.C. Liang, S. Foufou, S. Rachuri, C. Bock, and M. Mani, “Towards an ontology for open assembly model”, International Conference on Product Lifecycle Management, 2007, pp. 445–456.
[8] L. Louis-Sidney, V. Cheutet, S. Lamouri, O. Puron, and A. Mezzai, “A conceptual model for the implementation of an inter-knowledge objects exchange system (ikoes) in automotive industry”, Journal of Engineering Applications of Artificial Intelligence 25(5), 2012, pp. 1090–1101.
[9] T. Lu, F. Guan, N. Gu, and F. Wang, “Semantic classification and query of engineering drawings in the shipbuilding industry”, International Journal of Production research 46(9), 2008, pp. 2471–2483.
[10] A. Matsokis, and D. Kiritsis, “An ontology-based approach for product lifecycle management”, Computers in Industry 61, 2010, pp. 787–797.
[11] D. Cao and M. Fu, “Port-based ontology modeling to support product conceptualization”, Journal of Robotics and Computer-Integrated Manufacturing 27(3), 2010, pp. 646–656.
[12] H. Panetto, M. Dassisti, and A. Tursi, “Onto-pdm: product-driven ontology for product data management interoperability within manufacturing process environment”, Journal Advanced Engineering Informatics archive 26(2), 2012, pp. 334–348.
[13] N. Chungoora, R. Young, G. Gunendran, C. Palmer, Z. Usman, N. Anjum, A. Cutting-Decelle, J. Harding, and K. Case, “A model-driven ontology approach for manufacturing interoperability and knowledge sharing”, Computers in Industry 64(4), 2013, pp. 392–401.
[14] M. Imran, and B. Young, “The application of common logic based formal ontologies to assembly knowledge sharing”, Journal of Intelligent Manufacturing, 2013.
[15] M. Lanz, T. Kallela, E. Jarvenpaa, and R. Tuokko, “Ontologies as an interface between different design support systems”, 9th WSEAS International Conference on NEURAL NETWORKS, NN08, Sofia, Bulgaria, 2008, pp. 202–207.
[16] R. Woll, C. Geiler, and H. Hakya, “Modular ontology design for semantic data integration”, 5th International Conference on Experiments/Process/System Modeling/Simulation/Optimization, 2013.
[17] Y. Zhang, X. Luo, H. Zhang, and J.W. Sutherland, “A knowledge representation for unit manufacturing processes”, International Journal of Advanced Manufacturing Technology, 2014, pp. 1–21.
[18] S.C. Brandt, A. Bernard, J. Morbach, M. Miatidis, M. Theienb, M. Jarke, and W. Marquardt, “An ontology-based approach to knowledge management in design processes”, Computers & Chemical Engineering 32, 2008, pp. 320–342.
[19] A. Felic, B. Knig-Riesa, and M. Kleinb, “Process-oriented semantic knowledge management in product lifecycle management”, 8th International Conference on Digital Enterprise Technology, 2014, pp. 361–368.
[20] D. Antonelli, G. Bruno, A. Schwichtenberg, and A. Villa, “Full exploitation of product lifecycle management by integrating static and dynamic viewpoints”, Advances in Production Management Systems, Competitive Manufacturing for Innovative Products and Services, 2012, pp. 176–183.

[21] G. Bruno, D. Antonelli, R. Korf, J. Lentes, and N. Zimmermann, “Exploitation of a semantic platform to store and reuse plm knowledge”, Advances in Production Management Systems - Innovative and Knowledge-Based Production Management in a Global-Local World 438, 2014, pp. 59–66.
[22] G. Bruno, J. Lentes, and A. Villa, “Design of a manufacturing knowledge management system”, Recent Advances in Electrical and Electronic Engineering, 2014, pp. 216–225.
[23] M. Fowler, and K. Scott, “UML distilled”, Addison-Wesley, 2000.
[24] M. Groover, “Fundamentals of modern manufacturing: materials processes, and systems”, John Wiley & Son, 2007
[25] Jena, 2014: https://jena.apache.org .
[26] Protégé, 2014: http://protege.stanford.edu.