International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:2, No:6, 2008

Effective Collaboration in Product Development via a Common Sharable Ontology

Sihem Mostefai, Abdelaziz Bouras, and Mohamed Batouche

***Abstract*—**To achieve competitive advantage nowadays, most of the industrial companies are considering that success is sustained to great product development. That is to manage the product throughout its entire lifetime ranging from design, manufacture, operation and destruction. Achieving this goal requires a tight collaboration between partners from a wide variety of domains, resulting in various product data types and formats, as well as different software tools. So far, the lack of a meaningful unified representation for product data semantics has slowed down efficient product development. This paper proposes an ontology based approach to enable such semantic interoperability. Generic and extendible product ontology is described, gathering main concepts pertaining to the mechanical field and the relations that hold among them. The ontology is not exhaustive; nevertheless, it shows that such a unified representation is possible and easily exploitable. This is illustrated thru a case study with an example product and some semantic requests to which the ontology responds quite easily. The study proves the efficiency of ontologies as a support to product data exchange and information sharing, especially in product development environments where collaboration is not just a choice but a mandatory prerequisite.

***Keywords*—**Information exchange, product lifecycle management, product ontology, semantic interoperability.

I. Introduction

PRODUCT development constitutes the core activity of many industrial companies. It includes actually a variety of business processes associated with the activities appearing in the product lifecycle. This is where PLM (Product Lifecycle Management) comes in. PLM is the business activity of managing an organization's products all the way across their lifecycles in the most effective way [1]. A PLM system can be described as "an enterprise-wide Information Technology (IT) infrastructure to support management of product definition throughout its complete lifecycle" (from initial concept to product obsolescence) [2].

PLM requires a holistic approach melding product-related application systems, data, processes, techniques and skills. Consequently, one of the major problems in this field is the vast amount of information that is available and the ability to make sense of it. There is an urgent need to an integration solution that allows correct access to relevant product information. This problem is related to interoperability issues and how they are addressed. Many techniques have been developed to tackle these types of heterogeneity at different levels: system, syntactic, structural and semantic levels. As product knowledge is gaining growing attention as a means to enhance productivity and performance in developing, manufacturing and selling customized products, our solution to the integration problem will be at the semantic level, via a common shared ontology. This latter is based on the idea of a common knowledge or a common semantic shared by different lifecycle phases having each a special perception or view of the product.

The remainder of this paper is organized as follows: section 2 is concerned with the issue of knowledge and semantics in the product development process. Section 3 gives an overview of the ontology concept and the ontology building process. Section 4 is dedicated to the description of the product ontology. In section 5, an example product is studied and used for instantiating the ontology. The ontology instantiation is then used to show how an effective collaboration is achieved. The paper is ended with some concluding remarks and important perspectives in section 6.

II. Knowledge in Product Development

Knowledge is considered as "the source" of innovation and growth for the enterprises involved in product development. A critical issue in this context is how to model, manage and utilize the product knowledge across the entire product lifecycle effectively and efficiently. PLM is an interesting perspective for addressing such questions. In concert with collaborative engineering, PLM aims to provide support for a broad range of business activities from the conception of a product to its disposal. Successful PLM oriented knowledge management should be undertaken in a holistic and multi-disciplinary perspective. All activities such as design, manufacturing and marketing affect product development right from the stage of conceptual design. Thus, a product development activity requires the expertise and interaction of a broad range of disciplines. Consequently a broad spectrum of knowledge is used and shared in these distributed teams. The coordination and integration of relevant product information throughout the whole company become critical issues. This problem has been addressed in many studies and is still gaining growing attention from researchers. In [3], the authors stress the necessity for companies to have a unified

Manuscript received November 14, 2005.

S. Mostefai and M. Batouche are with the LIRE laboratory, Mentouri University, Route de Ain El Bey, Constantine 25000, Algeria (phone: +21331614346; fax: +21331639010; e-mail: s.mostefai@wanadoo.dz and batouche@wissal.dz).

A. Bouras is with CERRAL center, Campus Porte des Alpes, 160, Boulevard de l'Université, 69676 Bron Cedex, Lyon, France (phone : 33(0)478773146; fax: 33(0)478006328; e-mail: abouras@univ-lyon2.fr).

view on product knowledge in order to better support and synchronize concurrent activities. Enterprises are moving away from unilateral, locally optimized views on product data toward unified product models that foster editing of and access to product information for employees, partners, and customers throughout all business processes. The authors suggest a product information architecture that addresses the topic of organizing product information in a structured way and at different levels starting at the syntactic level, followed by the data level and ending with the semantic level.

Considering the large amount of information associated to the product development process, ant noticing that the shape represents an important part of it, some authors have tried to derive the semantic from the shape [4]. Observing that a successful information integration at the semantic level could not be achieved without an efficient knowledge sharing strategy, many authors in artificial intelligence have developed techniques for knowledge capturing and representation in order to build sharable knowledge bases in various fields[5] as well as in collaborative product development [6], leading to ontological approaches developed subsequently[7], [8]. Ontologies are considered as the most recent knowledge representation models.

III. Ontologies: A General Survey

A. What is Ontology?

The term ontology is borrowed from philosophy. Merriam-Webster dictionary defines it as "*a branch of metaphysics concerned with the nature and relations of being* "[9]. In artificial intelligence, what exists is that which can be represented. *Ontology is an explicit specification of a conceptualisation* [10]. A conceptualisation is the set of objects, concepts and other entities that are assumed to exist in some area of interest together with the relationships that hold among them. A *conceptualisation is an abstract simplified view of the world that we wish to represent for some purpose* [10]. Regardless of the domain, an ontology consists of several components, the most important are: *concepts*, *relations*, *attributes*, *instances* and *axioms* [10].

Ontologies are gaining great attention in various disciplines, especially in fields where knowledge has a structured nature. This makes it easy to extract the relevant concepts, relations, attributes and axioms used in the ontology. Once this is done, the ontology undergoes a series of refinement steps to become machine exploitable. Interoperating and information sharing thru a well organized controlled vocabulary; indexing complex information and combat combinatorial explosions are among the most representative roles of ontologies.

B. Ontology Building Process

Building a common shared understandable ontology in any domain of interest constitutes a big challenge in itself. A range of methods and techniques for ontology building have been reported in the literature. Ushold's methodology [11], Grüninger and Fox's[12] and Methontodology [13], [14], which constitutes an excellent review about ontology building strategies, and many other variants[15], [16], [17] are among the most representative. Practically, all these techniques converge in defining the following general steps for the ontology building process:

1) *ontology capture*: is the identification and definition of key concepts and relationships in the domain of interest and the terms that refer to such concepts.
2) *ontology coding*: deals with formalizing such definitions and relationships in some formal language.
3) *ontology integration*: deals with associating key concepts and terms in the ontology with concepts and terms of other ontologies; that is, incorporating concepts and terms from other domains.

The first step is generally begun by a knowledge abstraction phase whose role is the identification of the relevant pieces of knowledge in a domain. These pieces of knowledge are submitted to a series of refinements in subsequent steps in order to make them more formal and to derive the concepts, relations and attributes of the corresponding ontology.

We have chosen to use the Protégé2000 ontology development environment and its OWL plugin, following the framework described in [17]. The choice of OWL is justified by the fact that it is so far the most recent development in standard ontology languages from the World Wide Web Consortium (W3C)[19]. Like Protégé, OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators - e.g. and, or and negation. Furthermore, it is based on a different logical model that allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognise which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly.

IV. Description of the Product Ontology

The product ontology we are suggesting here is based on three selected views, namely: part detail design view, assembly view and manufacturing process planning view. These sample views have been selected mainly because they exhibit many common semantic aspects.

A. Role of the Feature Concept

The concept of feature appeared firstly in the field product engineering where it was defined as: "a representation of shape aspects of a product that can be mapped to a generic shape which is functionally significant for some product lifecycle phase" [18]. Originally, it was tightly linked to the product geometry. However, as product development does not include only engineering activities, product information was not merely restricted to geometry, it holds indeed a richer and more complex semantic content (functional, structural, behavioural, technological…). In order to capture this semantic in our approach, the meaning of feature has been extended to have a relevant definition according to the context it is used in, thus bridging the gap between geometry and other product information. The feature concept is intended to play an important role in the product ontology.

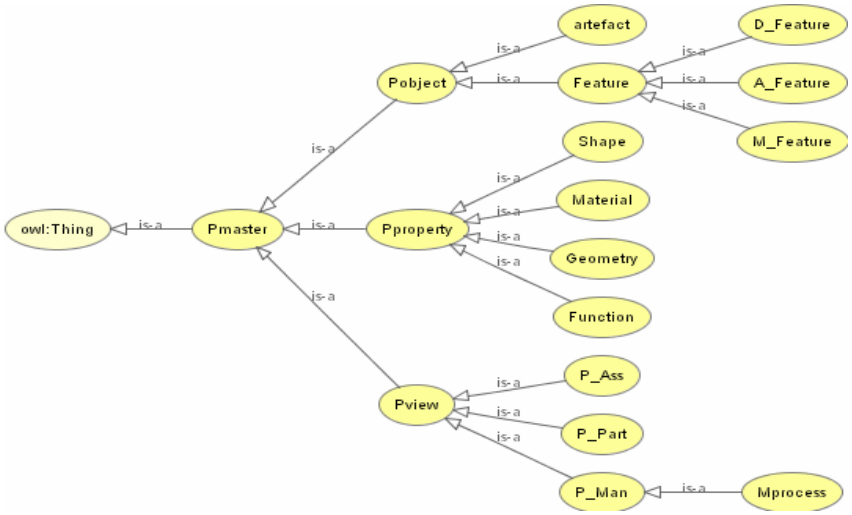

Fig. 1 Concept hierarchy of the product ontology

B. Product Ontology

The product ontology is described by a concept hierarchy (often termed taxonomy) showing the main concepts or classes, and a description of class associations. In the following, the terms *class* and *concept* are used indifferently. In order to make the ontology as robust as possible, the description has been limited to generic product attributes. Domain specific attributes have been intentionally omitted. The concept hierarchy is shown in Fig. 1.

At the highest level of the class hierarchy is the product master model class: *Pmaster* which is an abstract class from which all other classes are specialized. The common attributes: *name*, *type* and *generic_info* for all product classes are defined in this class.

At the second level of the class hierarchy are three different product categories, which represent specializations of the *Pmaster* class. We have three subclasses: Pproperty, Pview and Pobject, as described below:

- *Pproperty*: is an abstract class which is intended to describe some important product properties. Concrete specializations of this class are: *Material, Function, Form* and *Geometry* classes.
- *Pview*: is an abstract class used to describe different product views. It is specialized into three classes:
  - *P _Part*: which is intended to describe products consisting of a single part. A part is composed of a single *Material* and is supposed to fulfil a certain *Function*. This is represented by the *has_material* and *has_Pfunction* associations respectively.
  - *P_Ass*: describes more complex products constituted of part assemblies, As generally mechanical products consist rarely of a single part, this class describes many product categories. The product is seen as a set of components and interfaces between them. A component can be compound or simple. Compound components represent new sub-assemblies. Simple components are single parts. This information is visible thru two aggregation links: a containment relationship *sub-assembly/sub-assembly-of* defined on the *P_Ass* class itself and a *consists-of* association between *P_Ass* and *P_Part* meaning that an assembly is constituted of single parts,
  - *P_Man*: describes the manufacturing process planning view which is intended to map the design of the product onto the methods used to create it. Process planning involves: recognizing the relevant elements of the product that are of interest for the manufacturing task, namely: machining or manufacturing features. A manufacturing feature is commonly defined as a collection of related geometric elements that correspond to a particular manufacturing method or process, or which can be used to reason about a suitable manufacturing method or process for creating that geometry.
- *Pobject*: stands for product object class, this is an abstract class that has two subclasses:
  - *Feature*: is the feature class of the product. This class is generic and may have different meanings, depending on the product view. Therefore, it has got three sub classes corresponding respectively to the three product views described before, namely:

D_Feature class which describes the design features present on the part view, *M_Feature* class describing the manufacturing features of the manufacturing view, and *A_Feature* class which describes the assembly features existing in an assembly view. Whatever the view, the *Feature* class is *associated* to a *Function and a Shape* via *has*_Ffunction and *has_shape* associations. Further, a *Shape* is realized by a *Geometry*.

- *Artifact*: refers to a product or one of its components.

V. Example Product: The Vacuum Suitcase

We have chosen to instantiate the product ontology with an example assembly: the vacuum suitcase. This product is used to enable silicon wafers to be processed by multiple facilities.The example is available at the Drexel University repository http://www.designrepository.org/. It should be noted that despite its relative simplicity, the product descriptions mentioned in this paper assume certain knowledge of mechanical product design. Providing detailed and complete definitions is out of the scope of this paper. The main goal is to show how the ontology is instantiated and exploited for collaborative goals.

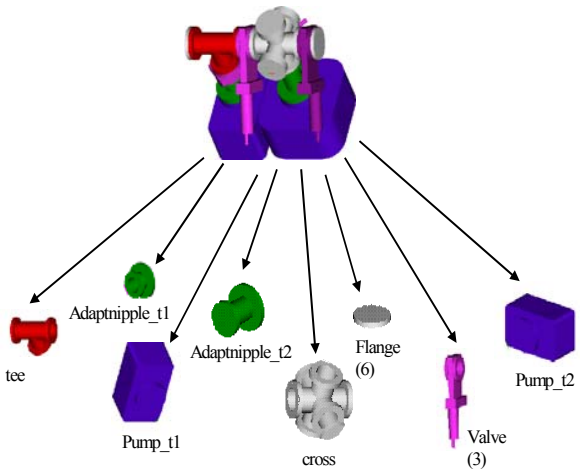

Fig.2 Solid model and parts of the vacuum suitcase

A. Technical Description

A Vacuum Suitcase is a prototype for the transport of VLSI chips among different fabrication facilities. It consists off fifteen parts as shown in Fig. 2.

The list of parts and their functional description is given in Table I.

B. Assembly Hierarchy

The vacuum suitcase system is composed of two sub-assemblies as shown in Fig. 3. The two sub-assemblies are: (1) *system_puimp1* assembly containing the following parts: pump_t1, adaptnipple_t1, flange, tee and two valves; (2) *system_pump2* assembly that consists of: pump_t2, adaptnipple_t2, cross, five flanges and valve.

TABLE I
Part List of the Vacuum Suitcase

Name	Quantity	Functional Description
Tee	1	Connects pump_t1 with cross via flange.
Adaptnipple_t1	1	Adjusts valve with pump_t1.
Adaptnipple_t2	1	Adjusts valve with pump_t2.
Flange	6	Keeps the components in place and covers tee and cross to secure the whole system.
Cross	1	Connects pump_t2 with cross via flanges
Pump_t1	1	Ensures moving liquid or gas inside the vacuum suitcase.
Pump_t2	1	Ensures moving liquid or gas outside the vacuum suitcase.
Valve	3	Opens or closes openings to allow or prevent the flow of liquid or gas to and

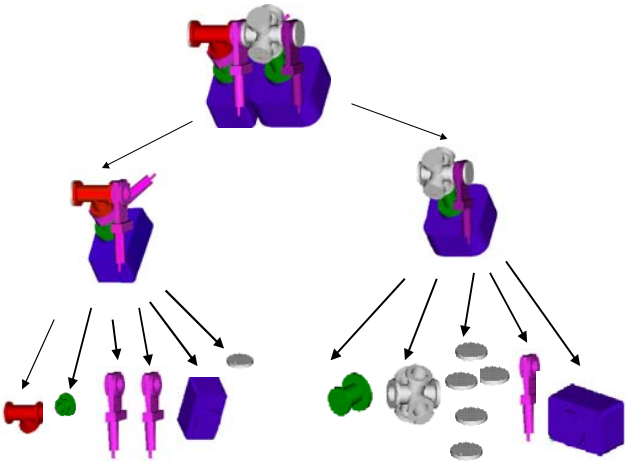

Fig. 3 Assembly hierarchy of the vacuum suitcase

C. Example Description in Potégé-OWL

We describe herein how the vacuum suitcase system is implemented in Protégé-OWL mainly by screenshots showing the ontology and its instantiation. Fig. 4 shows an instance tree of the vacuum suitcase system, the left side contains the class hierarchy of the Product_ontology, the middle side shows fifteen instances of the *P_Part* class corresponding to the vacuum suitcase parts, the right side shows some details of the *P_Part* class structure.

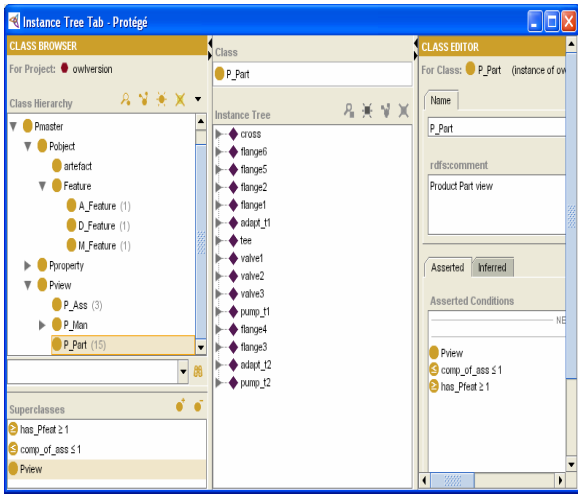

Fig. 4 Vacuum suitcase implementation in Protégé-OWL

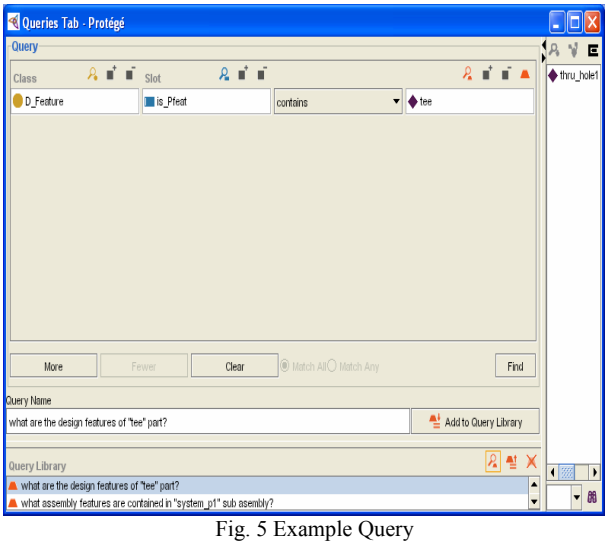

Fig. 5 Example Query

D. Enhancing Collaborative Work

Enhancing collaborative work is the main objective of the product ontology. This is illustrated in several ways: query mechanism, ontology browsing and ontology inference.

1. Query Mechanism

In the Protégé platform, it is possible to make queries on the ontology once it has been instantiated. The queries can be stored in a query library from which they can be retrieved and executed whenever needed. We have tested some typical queries:

- What assembly instances are sub-assemblies of "vacuum suitcase" assembly?
- What part instances compose "system_p1" assembly?
- What are the design features of the "tee" part?

Fig. 5 shows a query screenshot: at the top of this screen are the classes and properties used in the query, at the bottom, the query name and some of the queries stored in the query library, the right side is used to show the query results. The query tested in this example is*: What are the design features of the "tee" part?*

The queries can be more or less complex. Moreover, these queries can be easily handled by conventional database mechanisms. Nevertheless, they constitute a first step toward an effective collaboration. It is interesting at this stage to gather a maximum number of queries of different kinds and to categorize them in order to create specialized query libraries. Each library may concern a different product view.

2. Ontology Browsing

Very frequently, actors of different views and contexts may refer to equivalent concepts and properties without really knowing their equivalence. This situation costs a lot of time and money to product development organizations. When browsing the ontology, this equivalence is made explicit. This greatly contributes to enhance mutual understanding and removes linguistic and administrative barriers, thus, enabling more effective collaborative work.

Furthermore, using OWL as an ontology language contributes largely to the broad adoption of the ontology as it is expected to become a web standard. By the time this article was written, we have only exploited some of the important features of OWL, and indeed, it offers a large spectrum of interesting tools and characteristics we have not explored yet. These features might certainly contribute to support effective collaboration in the context of product development.

3. Ontology Inference

With OWL ontology language, it is possible to infer an ontology. That is to deduce new knowledge derived from the ontology content. In our case study, we show an example of such an inference deduced after a complex query. The query implies the collaboration of the part designers and the part manufacturing engineers. For instance, if the designers want to know what processes can be used in manufacturing a special feature on a part, say for example, design feature *thru_hole*, in order to fix the

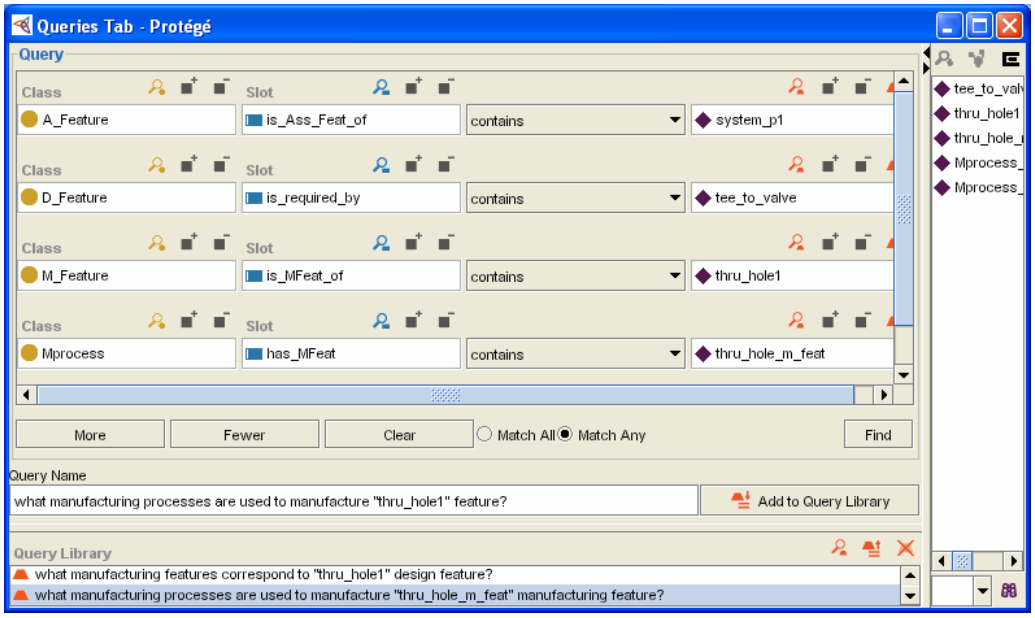

Fig. 6 Complex query example

manufacturing cost, the request necessitates their collaboration with the manufacturing engineers. Such information is not directly available in the product ontology, but it can be deduced after a complex query, as shown in Fig. 6. The query departs from the assembly features of *system_p1* sub assembly. Once these have been determined, the query proceeds by looking for the design features required by the assembly features, and among these, the *thru_hole* is explored to find what manufacturing features are used to obtain it. The manufacturing features are determined and denote two manufacturing processes (forging process or milling process). Thus, we can deduce that there are two manufacturing processes associated to the fabrication of the *thru_hole* feature. This deduction is possible, although not straightforward.
Such a possibility is the consequence some interesting features of OWL.

OWL ontologies may be categorised into three species or sub-languages: OWL-Lite, OWL-DL and OWL-Full. The version used in this work is OWL-DL which is based on Description Logics (hence the suffix DL). Description Logics are a decidable fragment of First Order Logic and are therefore amenable to automated reasoning. It is therefore possible to automatically compute the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL-DL. With such characteristics, OWL-DL ontologies are more formal. Thus, they can be used by software agents for performing tasks that were previously done by humans. In the product development field, this possibility is very interesting if well exploited. Especially in collaborative work that implies different profiles of partners. A software agent reduces considerably time consuming tasks previously performed by human experts and ensures a transparent collaboration while processing the ontology.

VI. Conclusion and Discussion

This paper presented a new ontology-based solution to the problem of information integration in product development environments. The approach presented here exploits the idea of a common knowledge or common semantic shared by different product views. The study is mainly based on some example views taken from the mechanical product development field. The purpose of an explicit product modeling ontology is to facilitate a common understanding among different people such as engineers with their CAD/CAM experience, production planners, IT technicians, etc. If all these people agree to accept an ontology, they can contribute to a unified product model. The success and broad adoption of such an ontology depend directly on this consensus. To promote large acceptance of a unified product model, researchers are investigating normalisation issues. Interesting perspectives in this sense are studied at NIST(National Institute of Standards and Technology) [20], [21] [22] aimed at proposing a base-level product model that is open, non proprietary, generic and capable of capturing the full engineering context commonly shared in product development.

This study presents many advantages:

– The proposed ontology is extendable and does not need major modifications every time another view is added.

– The ontology has been restricted to generic information content avoiding specificities in order to make it open and easily adaptable to many contexts.

At present, our immediate objectives are:

– To complete and enrich the product ontology with more important semantic content, in order to be better exploited, especially by detecting for instance equivalent and disjoint concepts and properties.

– To submit it to a consistency checking mechanism in order to ensure its correctness.

– Explore other OWL features that allow a better and more effective collaboration.

REFERENCES

[1] D. Bourke, "The PLM Software Scene: Tips for Smart Shopping", *2PLM e-zine*, vol.. 6, no. 12, 2004.
[2] J. Portella, "Collaborative management of the product definition lifecycle for the 21st Century", *in CD-ROM Proc. PDT Europe Conference*, Noordwijk, 2000.
[3] P. Ackerman, and D. Eichelberg, "Product Knowledge Management", *in Proc. International Conference on Economic, Technical and Organizational Aspects on Product Configuration System,*. Technical University of Denmark, Copenhagen, 2004.
[4] G. Brunetti, and S. Grimm, "Feature ontologies for the explicit representation of shape semantic", *International Journal of Computer Applications in Technology 2005,* vol. 23, no.2, pp. 192—202, 2005.
[5] R. Fikes, M. Cutkosky, T. Gruber, and J. Van Baalen, "Knowledge sharing technology: project overview", *Technical Report KSL 91-71,* Stanford University, Knowledge Systems Laboratory, 1991.
[6] T.R. Gruber, J.M. Tenenbaum, and J.C. Weber, "Toward a knowledge medium for collaborative product development", in *Proc. Second International Conference on Artificial Intelligence in Design*, Pittsburgh, 1992, pp. 413-432.
[7] T.R Gruber, "The Role of common ontology in achieving sharable, reusable knowledge bases", in *Proc. Second International Conference on Principles of Knowledge Representation and Reasoning:*, Cambridge, MA, 1991, pp. 601-602.
[8] T.R Gruber, "Toward principles for the design of ontologies used for knowledge sharing", *in Proc. International Workshop on Formal Ontology*. Available as *Stanford Knowledge Systems Laboratory Report KSL-93-04,* 1993.
[9] Merriam-Webster dictionary. Available online at: http://m-w.com.
[10] T.R. Gruber, "A Translation approach to portable ontology specifications", *Knowledge Acquisition,* vol. 5, no. 2, pp.199-220, 1993.
[11] M. Uschold, and M. Grüninger, "Ontologies: principles, methods and applications", *Knowledge Engineering review,* vol. 11, no. 2, 1996.
[12] M. Grüninger, and M.S Fox, "Methodology for the design and evaluation of ontologies", *in Proc. IJCAI 95 on Basic Ontological Issues in Knowledge Sharing,* Monréal, 1995.
[13] M.F. Lopez, and A.G. Perez, "Overview and analysis of methodologies for building ontologies", *Knowledge Engineering review,* vol. 17, no. 2, pp.129—156, 2002.
[14] O. Corcho, M. Fernandez-Lopez, A. Gomez-Perez, and A. Lopez-Cima, "Building legal ontologies with METHONTOLOGY and WebODE" , 2004. Available online at: http://users.isoco.net/~ocorcho/documents/LawSemWeb2004_CorchoEtAl.pdf
[15] R. Prieto-Diaz, "A Faceted Approach to Building Ontologies", 2002. Available online at: https://users.cs.jmu.edu/prietorx/Public/publications/BulidOntologiesRPD-ER2002.doc
[16] M. Denny, "Ontology building: a survey of editing tools", 2002. Available online at: http://www.xml.com/pub/a/2002/11/06/ontologies.html
[17] N. Fridman Noy, and D. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology",2001. Available online at: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html
[18] W.F. Bronsvoort, et al "Essential Developments in Feature Modelling", in *Proc. CAD/Graphics'2001, seventh International Conference on Computer aided Design and Computer Graphics*, Kunming, China, 2001, pp.6--15.
[19] OWL web ontology language guide. W3c Recommendation. 10 Feb 2004.
[20] R. Sudarsan, et al. "Object oriented representation of electromechanical assemblies using UML", *NIST internal report NISTIR 7057*, 2003.
[21] R. Sudarsan et al., "Information models for product representation: core and assembly models", NIST internal report *NISTIR 7173,* 2004.
[22] E. Subrahmanian, et al., (2005) "Product Lifecycle Management support: a challenge in supporting product design and manufacturing in a networking economy", *in Proc. International conference on Produt Lifecycle Management PLM'05*, Lyon, France, july 2005, pp. 495-5006.

Sihem Mostefai is a lecturer in the Computing Science Department at Mentouri University of Constantine, Algeria. She received her MSc in computer graphics in 1993, in addition to an engineer Diploma in computer systems in 1989 from the same university. She is also preparing a PhD in product lifecycle management applied to the case of mechanical products. Her research interests are in industrial engineering, ontologies, feature-based modelling, and data exchange.

Abdelaziz Bouras is full professor in industrial engineering at IUT Lumière Lyon II, and head of CERRAL Center, at PRISMa Laboratory, Claude Bernard University, Lyon, France. He was in 1998 at the Manufacturing Engineering Laboratory (MEL) of National Institute of Standards and Technology (NIST) as a guest researcher. He received his PhD in computer science from Claude Bernard University in 1992 and his engineer Diploma in mechanical engineering in 1987 from The National Institute of Mechanical Engineering (INGM) of Boumerdas, Algeria. His research interests are in solid and feature based modeling, industrial engineering, collaborative design, and data exchange.

Prof. Bouras is editor in chief of the IJPLM (International Journal of Product Lifecycle Management) journal and co-founder of the PLM'05 international conference. He is also co-author of *Product Lifecycle Management: Emerging solutions and challenges for Global Networked Enterprise*, Geneva, Switzerland: Inderscience Enterprise Limited, 2005.

Mohamed Batouche received his MSc and PhD degrees in computer science from the Institut National Polytechnique de Lorraine, France, in 1989 and 1993, respectively. Currently, he is a full professor at the University Mentouri of Constantine, Algeria. His research areas include artificial intelligence and pattern recognition.