Semantic tags for generative multiview product breakdown

Thomas Paviot, Vincent Cheutet, Samir Lamouri

Semantic tags for generative multiview product breakdown

Thomas Paviot*, Vincent Cheutet and Samir Lamouri

Supméca/LISMMA,
3, rue Fernand Hainaut,
93407 Saint-Ouen, France
Fax: +33(1)0149453454 E-mail: thomas.paviot@supmeca.fr
E-mail: vincent.cheutet@supmeca.fr
E-mail: samir.lamouri@supmeca.fr
*Corresponding author

Abstract: The interoperability of IT systems that drive engineering and production processes (i.e. Product Data Management and Enterprise Resource Planning systems) is still an issue. The semantic meaning of product information has to be explicit in order to be able to exchange information between these systems. However, the product breakdown activity generates many disconnected product views over which the product semantics is disseminated and mostly implicit. This paper introduces a methodology allow describing an explicit product semantics independently from its components. It is based upon the concept of *tag* coming from the semantic and collaborative web. The transposition of this concept to the domain of product development requires a new concept enabling to generate any view according to the business viewpoint: the semantic group.

Keywords: interoperability; multiview; product breakdown; semantic tag.

Reference to this paper should be made as follows: Paviot T., Cheutet V. and Lamouri S. (2010) 'Semantic tags for generative product breakdown', *7th Int. Conf. Product Lifecycle Management.*

Biographical notes: Thomas Paviot is an Assistant Professor of Mechanical Engineering and Information Technology in the Department of Mechanical Engineering at the Supmeca engineering school, France. He's also currently a PhD candidate.
Dr. Vincent Cheutet is an Associate Professor of Mechanical Engineering and Information Technology in the Department of Mechanical Engineering at the Supmeca engineering school, France.
Dr. Samir Lamouri is a Professor of Production Systems in the Department of Mechanical Engineering at the Supmeca engineering school, France.

1 Introduction

The industrial context is characterized by the growing complexity of the product development environment: the organizations are becoming more and more complex due to the market dynamics and economical constraints (supplier of rank n, partner multiplication, merger/acquisition, etc.). At the same time, the products are becoming more and more complex due to the integration of multidisciplinary technologies. Tools and methods used in the product development process have to be robust to any perturbation from the economical environment. The concept of Extended Enterprise and the Product Lifecycle Management (PLM) strategy are solutions allowing to achieve objectives of cost/quality/timeliness in a complex framework (Bernard et al. (2009)). Enabling extended enterprise thus implies to be able to connect enterprise IT systems and make them interoperate. The context of this paper is related to interoperability issues in the PLM field, and especially focuses on the IT systems that drive engineering and production processes, i.e. Product Data Management (PDM) and Enterprise Resource Planning (ERP) systems (Paviot et al. (2009)). The main target of this research work is to achieve data consistency over heterogeneous systems in order to ensure they can interoperate. Interoperability here refers to the ability of a system or a process to use information and/or functionality of another system or process (Vernadat (2007)). The present paper studies the *product breakdown* activity and its output: the product structure also known as *product view* in the literature. The goal is to point out open scientific issues from an interoperability viewpoint, and to propose a solution to overcome them. Section 2 first defines and describes a state-of-art on product structure. Section 3 presents the scientific issue to overcome and the general constraints to satisfy for searching a relevant solution. In section 4, the concept of *tag*, coming from the semantic web, is then adapted to the domain of collaborative product development process. A data model is defined and implemented in section 5. Finally, section 6 opens new perspectives.

2 Product breakdown

The product breakdown is defined as a process allowing to manage product complexity by decomposing it in a hierarchical set constituted by atomic elements. The result of this process is the *product structure*. Having a product structure permits then to grasp a complex product and to efficiently collaborate by assigning decomposition elements to resources able to process them (designer, maintenance operator, purchaser, machine-tool, workcenter etc.).

2.1 Product structure in CAD/PDM/ERP softwares

Most of the Computer-Aided Design (CAD) software tools (e.g. Catia™ V5, Solidworks™, ProEngineer™) propose a hierarchical view of the product allowing to reach the expected point of view in the design process. Product Data Management (PDM) software tools (e.g. Enovia™, Windchill™ or TeamCenter™), also use a product structure composed of items. These structures are generally referred as *Engineering Bill Of Material* (EBOM) since they organize

data used and produced during the design or re-design activities. According to the point of view of these software editors, the semantics attached to the product hierarchy, and especially the structure links, is implicit: neither the tool documentations nor their normal usage give a hint in a unambiguous way how to identify the product components or how to organize them (in a functional way? Organic ? etc.). Each entity (people or organizations) has to specify a methodology (or a set of best practices) in order to organize the product according to an explicit and shared semantics covering the whole extended enterprise (Tomovic et al. (2009)).
The Enterprise Resource Planning (ERP) systems (e.g. Adonix^{TM}, SAP^{TM} or Baan^{TM}), used to support the manufacturing management processes, also require a product breakdown as an input to the Material Requirement Planning (MRP) module: this structure is called *Manufacturing Bill Of Material* (MBOM). The structure links appearing in the MBOM characterize a dependency in the product manufacturing process, and carry information on the quantity of necessary elements. Phantom items, specific elements of the MBOM in regard with the EBOM, aim at falicitating the production management for products with high diversity. Unlike the EBOM, the definition and understanding of the MBOM is shared since the definition proposed by Orlicky (1974): the granularity level of the breakdown is the element identified in an inventory. MBOM semantics is thus absolutely unambiguous: MRP modules of all ERP systems use the same definition for the concept of MBOM.

2.2 The product breakdown as the core of the STEP standard

The ISO 10303 standard (Pratt (2005)), known as *ST*andard for the *E*xchange of *P*roduct model data (STEP), handles the concept of *product structure.* Tursi et al. (2009) present and implement a STEP based product representation using the PDM Schema, whose breakdown semantics is implicit. Latest released Application Protocols (AP), e.g. AP239-Product Lifecycle Support, or still in-work (AP233-System Engineering), share the same basis for the hierarchical product representation. Specializations of the AP239 data model explicitly define a few breakdown semantics related to the product maintenance : functional, physical, systemic, zonal or hybrid views (ISO (2005)). Unfortunately, this precise semantics is strongly limited in scope and does not allow covering the whole product lifecyle.

2.3 Bill Of Material conversion and interoperability issues

Different approaches allowing to transfer product structures from one system to an other have been proposed in the literature. Paviot et al. (2008) propose a conversion from a CAD structure to a PDM structure. Xu et al. (2008) study the EBOM/MBOM issue, and define a set of transformation rules and an algorithm to automate the BOM transfer from one system to the other. Ou-Yang et al. (2003) develop a high-level PDM/MRP integration framework, in the sense that the proposed framework covers a huge functional spectrum: engineering change management, BOM transfer, inventory srap cost evaluation. The product model on which this solution relies is however not detailed. Methodologies presented in these papers are based upon an algorithm that automates a manual and repetitive activity. These algorithms can be performed only if the input structure (the one to transfer) complies to a well-known model. The asumptions for this model are restrictive thus this kind of approach can not be generalized to any product structure that can be met in engineering.

3 Product breakdown limitations and issues

3.1 The product multi-view issue

The previous section enlightens that the hierarchical representation of the same product depends on the actor's point of view: according to his/her needs, the designer, the manufacturer, the buyer or the maintenance operator will use the relevant product structure for his/her activity. Several views co-exist in a concurrent way for the same product, without having one of them that can be considered as the best or the master from the others. Thereby, on the contrary of the hypotheses of approaches previously described, some items of the EBOM may not appear in the MBOM (e.g. a function). Conversely, some items of the MBOM might not appear in the EBOM (e.g. raw material, phantom items). Finally some items can be instantiated in both EBOM and MBOM. As a consequence, the IT interoperability becomes difficult: the semantic seal of these systems prevents the data meaning to be understood by the systems to which they are transferred.

3.2 Semantics mixed to material of design/production information

Whatever the data models used for the product structure description are, one can find the following elements:

- tree nodes (e.g. items): they are the material of the product development process (design items, production ones, etc.)
- dependency links of type parent/child between nodes, the set links/nodes constituting a tree topology. These links carry a very strong but implicit semantics.

By considering the EBOM or the MBOM of a product, this product view gathers elements of different nature:

- the nodes or the material of the design or manufacturing processes (the items),
- the semantics coming from the metadata associated with the node. These metadata, generally classified at the same place, are however of a different nature, if it is a property assigned by the computer system (creation date of the item, file size of the document, weight) or resulting from a decision process (e.g. item bought or manufactured),
- the semantics resulting from the context of instantiation of the node: the same node can be either present in the EBOM or in MBOM,
- the semantics resulting from dependence links between nodes: this semantics is implicit if the nature of the dependence link has not been specified *a priori*.

The product structure can be so considered as not being the pertinent container to manage and organize the product information: the items or the material are spread over many databases (PDM or ERP) and the product semantics is spread in each product view.

3.3 Proposed approach

To solve the interoperability problem, the questions asked in section 3.1 and 3.2 have to be answered: is it possible to build a unique product view accepted by all actors of the product lifecycle? How to solve the problem of semantics spreading in different systems? How to comb out elements coming from the material of a process (e.g. the items) from information that give them a meaning? To simultaneously solve these three questions, the following approach is proposed: the atomic elements (items) and the carried information belong to separate and clearly defined universes: the items world and the semantics one. Information is carried on the items via the notion of tag. Finally, with this description, each product structure or view is generated on the fly. The next two sections precise these notions and the regenerating process of the product structure.

4 Collaborative web and semantic tagging

The notion of collaborative web appeared when the information technologies allowed each internet user moving from an information consumer to a shared information producer (Mathes (2004)). While thousands of people can populate a collaborative space with content, the pertinent organization and classification of this information becomes a strategic issue, otherwise the resources cannot be accessed. A preconfigured data organization is not suitable to handle millions of documents without any knowledge of their content *a priori*. In recent years, tagging systems, also known as *folksnomies*, have then become increasingly popular. These systems enable users to add keywords known as *tags* to Internet resources (e.g. web pages, images, videos etc.) without relying on a controlled vocabulary. Tagging systems have the potential to improve search, spam detection, reputation systems, and personal organization while introducing new modalities of social communication and opportunities for data mining (Marlow et al. (2006)). The social bookmarking website Delicious™ (http://delicious.com) is an example of this mutation. The contexts of collaborative web and complex product development can be considered as sharing analogies:

- both of them are complex frameworks for *collaborative* activites : many different actors can create/read/modify shared information according to their permission rights,
- both of them need a strong, well-structured and explicit data organization.

As a consequence, the objective of this study is to transpose the information organization (using *tags*) from the collaborative and semantic web field to the one of industrial product development in order to control its complexity.

4.1 The semantic tags

In the tagging process, one takes benefits to the participation of the internet to organize documents with the help of spontaneously used keywords known as *tags* (Marlow et al. (2006)). An internet user willing to share a picture of a monument of Paris will use the tag “picture”,“Paris” and “monuments” to describe this resource. The tags are then processed by the computer system as keys to answer queries from other internet users. By this tagging process, the user extracts by himself the semantics of the resources he/she wants to share, and he/she publishes as metadata of his/her document. Linked to the document, conveyer of meaning, the semantic tag will be the main object of our study.

4.2 A first data model

From this first definition, it is possible to build a simple data model for the product development process: each atomic element of the actor’s view (design item, manufacturing item, etc.) is called item. For each item, the user associates one or several semantic tags to describe the attached information from his/her view. In the class diagram (figure 1), the association relationship between an item and a tag is of type $*..*$: the semantics of each item can be described by several tags. In the same way, one tag can describe several different items (Note: in this paper, the **CamelCase** typesetting convention is used for class naming, private attributes are represented with an underscore before their name).

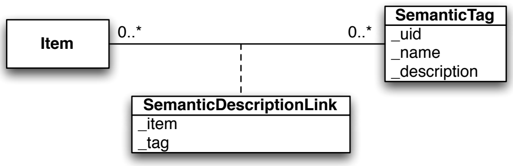

Figure 1 A first data model

This model faces three problems (Sen et al. (2006)):

- *typographical problem*: the tags work by spelling rapprochement. If one tag contains a spelling mistake or if a first user uses the plural form and another one the singular form, the system will not be able to make the rapprochement,
- *ambiguity problem*: for instance, the same acronym may be used to refer different resources,
- *flat structure problem*: the tags can be reconciled only if they describe common resources. Nevertheless, since the tagging process has for consequence to produce a large amount of tags, the users wish, from a set of tags, to directly organize them.

To solve the first two issues, the transfer of the tagging process to the product development process can be realized through a simple idea: each user has access, according to his/her role in the project, to an established corpus of semantic tags, from which he/she will choose the ones that seem adequate. The user liberty in the

tag choice is so constrained. Meanwhile, an organization problem arises from this solution: who will be in charge to constitute this corpus and, how?
On the other hand, the third problem is stated at the conceptual level. If no tag hierarchy is proposed, it will be impossible, from this description, to rebuild a product structure as a tree view. Two solutions can be considered (Bateman et al. (2007)):

- structured tags: the tags are separated by separators (e.g. slash, hash sign), for instance "Paris/Pictures/Monuments/Tour Eiffel" or "Paris#Cemetary#Père Lachaise". This approach does not suit the needs of our study,
- ontological tags: the semantic tags are linked to ontologies. Considered as inadequate for the collaborative web due to its ergonomic constraints, this approach takes advantage in our context to be semantically explicit and compatible with the corpus concept previously proposed.

The section 4.3 presents an adaptation of this concept for the product description: the *semantic group*.

4.3 Semantic group

A semantic group is defined as the ontological tag associated to a set of tags sets. The semantic group carries semantics appropriate for needs of a set of actors. Such groups can be for instance the *functional context* (that specifies the functions satisfied by the group of items), the *zonal context* (defining a volume or an area that contains the group of items), the *purchase context* etc. The previous list is not exhaustive: the semantic group can be whatever needed to represent the product according to the semantic domain. Figure 2 represents the UML class diagram modeling the semantic group, and the relations with tags and items. The semantic group is defined by the three following attributes: its unique identifier, name (e.g. *organic, functional, design_group 1* etc.) and description (in order to explicitly describe the semantics of the group). The relationship between an item and a semantic group can be of two types (figure 2):

- an instance of the **DefineGoup** class: the tagged item is a tree node in a specific context,
- an instance of the **InGroup** class: the tagged item is an element of the specified group. This relation carries one additional information: the quantity.

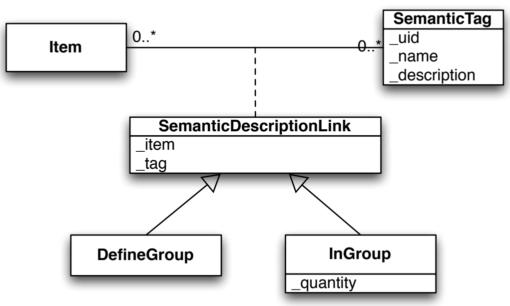

Figure 2 Extended data model taking the semantic groups into account

5 Example

To illustrate the approach proposed in the previous section, one can consider a simple product: a drilled metal sheet Sheet_1 is assembled with a drilled metal sheet Sheet_2 through the use of a bolt (i.e. the Screw_3 and the Nut_4). Each of the drilled metal sheets is obtained by a process of cutting/drilling from raw materials metal sheets Sheet_1_RM and Sheet_2_RM. The product is called *Assembly* in the following. Figure 3 proposes EBOM and MBOM for this product (these are just examples among all possible E/MBOM according to the design and manufacturing processes).

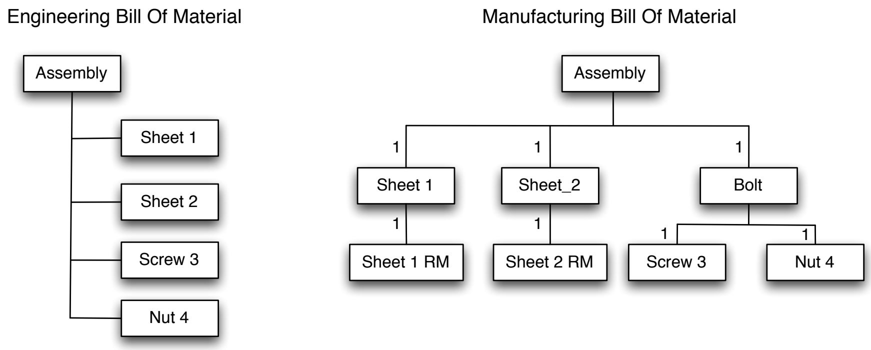

Figure 3 EBOM and MBOM for a simple example

The objective is to describe according to the previous model all the semantics available in this example. The item world is composed of the set of items appearing in the structures: Assembly, Sheet_1, Sheet_1_RM, Sheet_2, Sheet_2_RM, Screw_3, Nut_4 and Bolt. In the semantic space, it is necessary to create the following tags to describe completely information: Design, Manufacturing, DEsign Semantic Group 1 (DESG1), MAnufacturing Semantic Group 1 (MASG1), MASG2, MASG3 & MASG4. Figure 4 is a graphical representation of the links connecting items to relevant tags:

- the continuous lines represent relations of type **SemanticDescriptionLink**,
- the double lines represent relations of type is **DefineGroup**,
- the dashed line represent relations of type **InGroup**.

The transition from figure 3 to figure 4 enlightens that the two disconnected BOM have been merged into a *environment of a higher level*: the semantics of both EBOM & MBOM have been extracted and moved to an outer space, where explicit and complete semantics is described. As a result, there are no direct dependency links between items anymore, the connection between items is accomplished *via* the tags semantics.

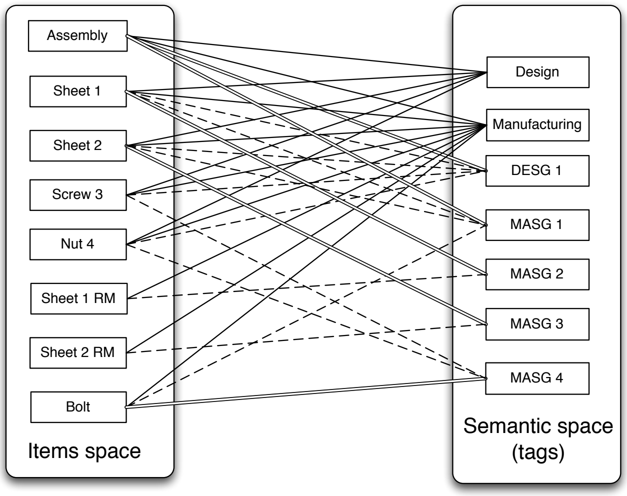

Figure 4 Original E/MBOM splitted into items and semantic spaces

The *higher level environment* presented above means that the BOMs can be inferred from the description of figure 4. From a simple algorithm looping over the list of items, it is obvious to get the ones matching a set of tags. The hierarchical treeview is generated from this description with the help of an algorithm that : a) gets all the items matching one semantic tag (e.g. Design); b) gets all elements which can be considered as tree nodes and c) gets all items being part of this group and are then children from the considered item. This algorithm has been successfully implemented in order to validate the example presented on figure 4.
This example demonstrated how to manage the reconcilialiation of two product breakdowns. Of course, the same method can be used to aggregate n breakdowns of a complex product, seen from n different viewpoints.

6 Conclusion - Further work

In this article, the product structure, as it is currently implemented and defined, has been presented as a difficulty when tackling the IT interoperability issue. The concept of hierarchical tags associated with the notion of semantic group has been proposed as a solution allowing to conceptually separate the items from their associated semantics. The so constituted semantic space is complete and consistent. The proposed concept should nevertheless be extended: a formal description of the ontology linked to the semantic group has to be realized, as well as its implementation based upon semantic web technologies enabling complex queries to be adressed to the framework.

References

Bernard, A., Laroche, F. and Da Cunha, C. (2009) 'Models and methods for knowledge formalisation in a PLM context', *3rd International Congress Design* *and Modelling of Mechanical Systems CMSM2009*, Hammamet, Tunisia.

Bateman, S., Brooks, C., McCalla, G. and Brusilovsky, P. (2007) 'Applying collaborative tagging to e-learning', *WWW*, Vol. 7.

ISO 10303-239 (2005) 'Industrial automation systems and integration - Product data representation and exchange - Part 239: Application protocol: Product life cycle support'.

Marlow, C., Naaman, M., Boyd, D. and Davis, M. (2006) 'Position paper, tagging, taxonomy, flickr, article, toread', *In Collaborative Web Tagging Workshop at WWW06*, pp.31–40.

Mathes, A. (2004) 'Folksonomies-cooperative classification and communication through shared metadata', *Computer Mediated Communication*, LIS590CMC.

Orlicky, J.A. (1974) 'Material requirements planning: the new way of life in production and inventory management', McGraw-Hill, Inc. New York, NY, USA.

Ou-Yang, C. and Cheng, M.C. (2003) 'Developing a PDM/MRP integration framework to evaluate the influence of engineering change on inventory scrap cost', *Int. J. Adv. Manuf. Technol.*, Issue 22, pp.161–174.

Paviot, T., Morenton, P., Cheutet, V. and Lamouri, S. (2008) 'MultiCAD/MultiPDM integration framework', *5th International Conference on Product Lifecycle Management*, Seoul, Korea.

Paviot, T., Cheutet, V. and Lamouri, S. (2009) 'Design and logistics IT federation through Product Lifecycle Support standard', *6th International Conference on Product Lifecycle Management*, Bath, UK.

Pratt, M.J. (2005) 'ISO 10303: the STEP standard for product data exchange, and its capabilities', *Int. J. Product Lifecycle Management*, Vol. 1, pp.86–94.

Sen, S., Lam, S.K., Rashid, A.M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F.M. and Riedl, J. (2006) 'Tagging, communities, vocabulary, evolution', *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pp.190–204.

Tomovic, C., Walton, A., Ncube, L., Grieves, M., Birtles, B. and Bednar, B. (2009) 'Measuring the Impact of Product Lifecycle Management: Process Plan, Waste Reduction and Innovations Conceptual Frameworks, and Logic Model for Developing Metrics', *Product Realization: A Comprehensive Approach*, ed. Springer US, pp.1–14.

Tursi, A., Panetto, H., Morel, G. and Dassisti, M. (2009) 'Ontological approach for products-centric information system interoperability in networked manufacturing enterprises', *Annual Reviews in Control*, Vol. 33, Issue 2, pp.238-245.

Vernadat, F.B. (2007) 'Interoperable enterprise systems: principles, concepts, and methods', *Annual Reviews in Control*, Vol. 31, No. 1, pp.137–145.

Xu, H.C., Xu, X.F. and He, T. (2008)'Research on Transformation Engineering BOM into Manufacturing BOM Based on BOP', *Applied Mechanics and Materials*, Vols. 10-12, pp.99–103.