



The Cone-BOM model for consistent and minimal product structures representation

Virginie Fortineau, Thomas Paviot, Samir Lamouri, Vincent Cheutet

► To cite this version:

Virginie Fortineau, Thomas Paviot, Samir Lamouri, Vincent Cheutet. The Cone-BOM model for consistent and minimal product structures representation. 8th International Conference on Product Lifecycle Management (PLM11), IFIP., 2011, Eindhoven, Netherlands. hal-00709038

HAL Id: hal-00709038

<https://hal.science/hal-00709038>

Submitted on 17 Jun 2012

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

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

The Cone-BOM model for consistent and minimal product structures representation

Virginie Fortineau*, Thomas Paviot,
Samir Lamouri and Vincent Cheutet

LCPI/Arts et Métiers ParisTech,
151, bd de l'Hôpital,
75013 Paris, France
E-mail: virginie.fortineau@ensam.eu
E-mail: thomas.paviot@ensam.eu
E-mail: samir.lamouri@ensam.eu
LISMMA/Supméca,
3, rue Fernand Hainaut,
93407 Saint-Ouen, France
E-mail: vincent.cheutet@supmeca.fr
*Corresponding author

Abstract: The lack of interoperability between information systems that support design and manufacturing is known as one of the major open issues in the field of Product Lifecycle Management. Especially, the BOM transfer between PDM and ERP suffers from a lack of consistent model that would enable a conservative propagation of updated information from one system to the other. This paper introduces a minimal model, based upon an ontological description, that aims at linking, in a consistent manner, design and manufacturing views. This model can be represented in a graphical way, using a cone topology to extend the semantics of the dependency links, leading to the so-called Cone-BOM model.

Keywords: interoperability; Engineering Bill Of Material (EBOM); Manufacturing Bill Of Material (MBOM); ontology; Description Logics (DL); Semantic Web Rule Language (SWRL).

Reference to this paper should be made as follows: Fortineau, V., Paviot, T., Lamouri, S. and Cheutet, V. (2011) 'The Cone-BOM model for consistent and minimal product structures representation', *8th Int. Conf. Product Lifecycle Management*.

1 Introduction

Product Lifecycle Management (PLM) is usually presented as a cross-cutting activity that goes from the design to the maintenance and finally to the product recycling stage [1]. Moreover, the strategic purpose of the PLM is to control the complexity of the dynamic system composed of a high number of interacting

subsystems such as suppliers, customers, employees, factories or heterogeneous informations systems involved in product development [2]. The latter plays a crucial role, since IT systems enable sharing/exchanging/distributing the product related information across the whole company. According to the Metcalaf's law, "the value of a network increases exponentially with the number of nodes (participants)" [3] under the condition that the level of interoperability increases at the same time. Interoperability is defined here as "the ability of two systems (or more) to communicate, cooperate and exchange services and data, thus despite the differences in languages, implementations, executive environments and abstraction models" [4]. Interoperability then appears as a major issue in a successful PLM deployment strategy. The litterature shows that this issue is still widely open, and that the lack of interoperability must be overcome [5].

According to [6], interoperability must be achieved at three different levels: the technical, semantic and organizational levels. The semantic level, which addresses the *meaning* of the data, is the focus of the present paper: this research deals with the way to preserve the semantic flow from any information loss during data transfer between design and production information systems, such as Product Data Management (PDM) and Enterprise Resource Planning (ERP) systems. Indeed, Ben Kheder et al. [7] state that "the main need today [in the production information systems] is the communication of updated data from engineering to production management".

The focus of this study is the EBOM/MBOM consistency issue for the purpose of sharing and exchange. The paper is structured as follows: section 2 introduces a state-of-the-art related to product models and their limitations and shows the contribution of ontologies in a federative approach of interoperability. Section 3 proposes a new minimal product model, based on an ontological description, suitable for the interoperability in the design/manufacturing interface. This model is illustrated on a case study in section 4, whereas section 5 discusses this model. Finally, section 6 concludes this paper and introduces perspectives for future works.

2 Product modeling at the design/production interface

The semantic interoperability can be achieved from several approaches [8]: integration (all the participants must use a shared data standard), unification (establishment of mapping rules between participants) and federation (dynamic collaboration of different participants). Many recent works have dealt with the unification approach [9] [10] [11] for semantic interoperability, known to be more agile and flexible than integration. Authors use either *ad hoc* data models or standard data models to perform data mappings that support information flow. Product data models used for unification provide a *static* description of product structures [12]. Additionally, the mappings are built manually, leading to static mapping tables, that are not totally compliant with highly agile and flexible solutions. The federative approach, based upon ontologies and reasoning, then seems interesting to overcome this limitation.

According to [13], ontologies are a "formal, explicit specification of a shared conceptualization". As [14] explained, "a conceptualization is the extraction of

vocabularies from a domain and is an abstract, simplified view of the world that we wish to represent”. The Description Logics (DL) are the best suited languages to express an ontology [15], and specifically the OWL languages. Between the three variants of OWL (Lite, DL and Full), we chose OWL-DL because it is the best compromise between expressiveness and reasoning [16].

Recently, several works in the PLM field are using ontologies [17] [18] [14] [9], concerning product modeling, process modeling and service modeling as well. Based on the literature review, following choices were made for the model we aim to build: a simple and dynamic data model, intended to product structures exchange, must be designed using ontologies and rules in order to perform reasoning and infer suitable mappings between information. The next section introduces this model.

3 A minimal product model based upon ontologies

The proposed conceptualization aims to extract semantics from the local (design and manufacturing in this case) views and to implement them in a higher-level minimal model, based on the use of ontologies. Then, by reasoning on the model, it is possible to extract inferred data and to resite them into the local views. In this way, the mapping between design and manufacturing element are indirectly but completely achieved. The following subsections present classes, properties and restrictions of this ontology.

3.1 EBOM/MBOM similarities and differences analysis

The EBOM and MBOM contain the various entities composing the product structure, but also the parent/child relationships between entities [19]. Although both BOMs share a common representation (*i.e.* a tree-like structure), they deal with very different meanings [20]: the EBOM defines the virtual product whereas MBOM describes a process view enabling the Material Requirement Planning [21] required for the manufacturing stage. The first step of the modeling process is to figure out similarities (and differences) between EBOM and MBOM:

- they both present a tree structure composed of nodes (items) and parent/child relationships. These relationships hold a large amount of semantics, but mostly in an implicit way [22],
- the nodes refer to items of different natures. EBOM items may not be present in the MBOM and *vice versa*. However, they refer to a specific meaning for the designer or manufacturer. As a consequence, all items have the same existing need and will be treated in the same way in the proposed model,
- in both BOM, the parent/child dependency links express a composition link. The parent is indeed not a new or separated element, but the addition (or union) of all its children,
- in both BOM, it is possible to explicitly state a *contact* link between the children. If the link between two children is not constant over the time, there do exist indeed a contact loop between all the children. That means that

between all children of the same entity, it is always possible to find at least one set of these children that are physically in contact. Moreover, each child is part of at least one of these sets. This contact link will be the model for *interactions* between both BOMs and is represented as a loop in the graphical representation of the proposed model, section 3.6.

The previous points drive the construction of the ontology: following subsections describe classes, properties, restrictions and domain specific rules of this ontology.

3.2 Description of the classes and properties

The ontology is described using the Protégé environnement [23], that uses OWL-DL language; and is represented on figure 1. The first class we create

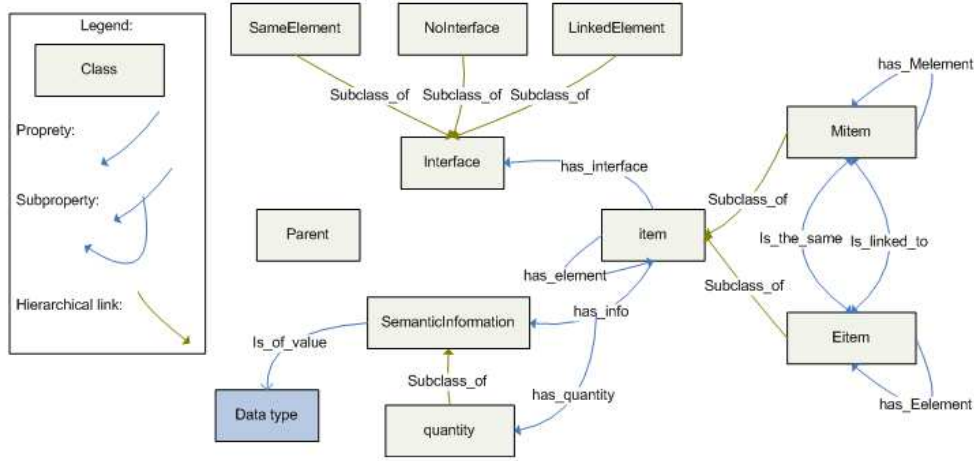


Figure 1 Description of the classes and properties of the ontology

is the **Item** class. Dealing with the design/manufacturing interface implies two subclasses: **Eitem** for the EBOM entities and **Mitem** for the MBOM entities. As this ontology aims at describing the link between MBOM and EBOM, the different **Item** have three possible configurations:

- the **SameElement** class corresponds to the items that exist in both BOMs,
- the **LinkedElement** class corresponds to the items that have a link with the other BOM,
- the **NoInterfaceElement** class corresponds to the items that refer to no item in the other BOM.

These three classes are subclasses of **Interface**. One of the queries that the ontology will need to answer is to find the isolated items (i.e. the item that belongs to the **NoInterface** class and have no children). These items have no reason to exist (the only logical **NoInterface** items are the phantom items or the

design assembly items, that do have children) and need a *is_linked_to* property. The ontology needs a class to express the existence (or not) of a parent/child link between items: the **Parent** class. Finally, as the model aims at providing all the available information hold by the items, the **SemanticInformation** class is created. This class may have as many subclasses as needed -like the **quantity** that represents the cardinality of the BOM- depending on the application (for instance, a material class in the steel industry, a color class in the jewelry industry, etc.). All the instances of these classes are data type.

3.3 Description of the properties

The object properties allow us to characterize and relate the different classes to each other. First of all, the *has_element* property is the expression of the parent/child relation that links items in each BOM. *has_Eelement* refers to the EBOM, as *has_Melement* refers to the MBOM. This property is *transitive* type and has logically the reverse property *is_element*. The property *has_element* refers to the class **Item**. In order to link the **Eitem** and **Mitem**, two properties are created :

- if the two connected items directly correspond to one another, the *is_the_same* property is used,
- if the two connected items are not exactly the same concept but influence one another, the *is_linked_to* property is instead used.

As it is more restrictive than the *is_linked_to* property, the *is_the_same* property is a subproperty thereof. The property *has_interface* expresses that a **Item** belongs to an **Interface** class.

Finally, the data properties link the **SemanticInformation** to the corresponding **Item** instance. *has_info*, that is the generic data property used that for, may have diverse subproperties to link the different kinds of information to the instances, such as *has_quantity* that expresses the cardinality of the BOMs.

3.4 Restrictions

To ensure the consistency of the ontology, restrictions are added:

- *has_element* links only two individuals of the same **Item** subclass,
- *is_linked_to* and *is_the_same* link two individuals of different **Item** subclasses,
- individuals of the **Item** class have at most one *is_the_same* property and exactly one *has_interface* property.

3.5 Domain specific rules

The OWL-DL reasoning capabilities are not sufficient to express all the requirements of product modeling [18]. Domain specific rules are added using SWRL language (unification of OWL-DL and RuleML). A way to implement

SWRL in the Protégé environment is to use SWRLTab, executed by Jess. To complete product modeling, the following domain specific rules are defined:

- an **Item** that has the *is_linked_to* property also belongs to the **LinkedTo** class, and an **Item** that has the *is_the_same* property also belongs to the **SameElement** class.
- An **Item** that belongs neither to the **SameElement** nor to the **LinkedTo** classes belongs to the **NoInterface** class,
- An **Item** that have the *has_element* property belongs to the **Parent** class.

3.6 Graphical representation of the model

In order to make this model more understandable, a graphical representation is proposed (see figure 2). The discussion related to EBOM and MBOM suggests that the geometrical representation of an item should have two aspects: one loop and a top. The cone topology thus appears as a relevant choice.

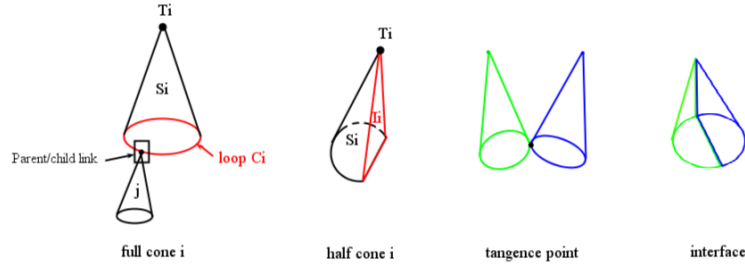


Figure 2 A graphical 3d view of the model

A cone i is composed of a top T_i , a surface S_i and a circular base C_i . The top and the surface represent the item i , as the loop C_i is the locus of junction with the children. The surface enables to store semantic information about the item. We also define a half cone, that presents an interface surface I_i (I_i is geometrically a infinite plane). I_i is the interface surface between the design and the manufacturing corresponding cones. Thanks to the reasoning, it is possible to edit a design/manufacturing view of the assembly. The different steps to create the graphical view of the assembly from the ontology are as follows:

- the retrieving capabilities of DLs are used to find all instances of the **SameElement** class: a half-cone is created for each resulting instance. Similarly, a full cone is created for each element of the **LinkedTo** and **NoInterface** classes,
- individuals of the **Eitem** class are colored in blue, whereas individuals of the **Mitem** classe are green,
- for each cone created, we add geometrical constraints as follows: if i *is_element* of j , then $T_i \in C_j$. For the **LinkedTo** individuals, we create

the tangence point $P \in C$. If i *is_linked_to* j , then $P_i = P_j$. For the *SameElement* individuals, if i *is_the_same* j , then $I_i = I_j$ and $T_i = T_j$,

- the *SemanticInformation* of each *Item* i are stored on the surface S_i .

4 Case study

4.1 Description of the case study

The case study is a simplified view of a car assembly. Two possible EBOM and MBOM for this simple case are presented on figure 3.

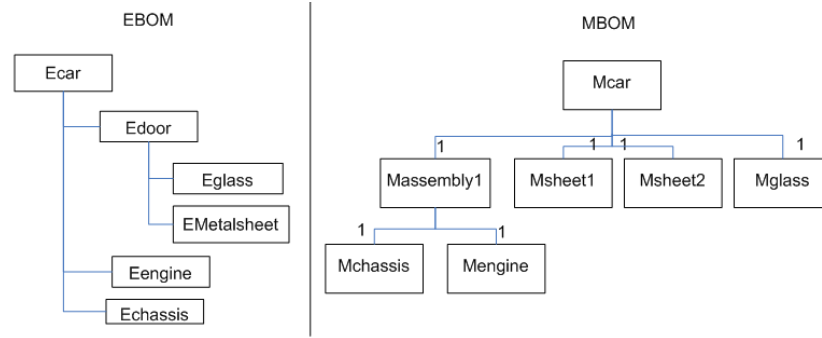


Figure 3 BOM of the case study

From the designer viewpoint, the car is composed of an **engine**, a **door**, and a **chassis**. The **door** is itself composed of a **glass** and a **metal sheet**. For the manufacturer, the car is produced by assembling the **engine** and the **chassis** to create the **assembly₁** item. Then the **assembly₁**, the **glass** and the two **metal sheets** are assembled together. All the items of the EBOM are declared as individuals of the *Eitem* class. Similarly, all the item of the MBOM are declared as individuals of the *Mitem* class. Their names spelling start with “E” or “M”.

4.2 Ontological description and modelisation of the case study

The following mapping are expressed manually in the ontology:

- **Ecar** *hasElement* **Echassis**, **Eengine** and **Edoor**, **Mcar** *hasMelement* **Massembly₁**, **Mglass**, **Msheet2** and **Msheet1**, **Edoor** *hasElement* **Eglass** and **EMetalsheet**, **Massembly₁** *hasMelement* **Mengine** and **Mchassis**,
- **Ecar** *is_the_same* **Mcar**, **Eglass** *is_the_same* **Mglass**, **Echassis** *is_the_same* **Mchassis**, **Eengine** *is_the_same* **Mengine**.

One of the reasoning capabilities of ontologies that can be used for this case study is to find the isolated items defined section 3.2. A first retrieving

action shows that the Item **Emetalsheet**, **Msheet1**, **Msheet2**, **Edoor** and **Massembly₁** belong to the **NoInterface** class. But with a second retrieving action, it appears that among these items, only **Emetalsheet**, **Msheet1** and **Msheet2** are not **Parent**. There is a lack of mapping on these items revealed, and we declare that the **Emetalsheet** *is_linked_to* the **Msheet1** and the **Msheet2**.

A 3d view of this model is presented on figure 4: it should be read as a 3d dimensional extension to the usual 2d view of EBOM and MBOM as they are presented in figure 3. The figure left side does not take account of the *is_linked_to* property added between the isolated items, whereas the figure right side does.

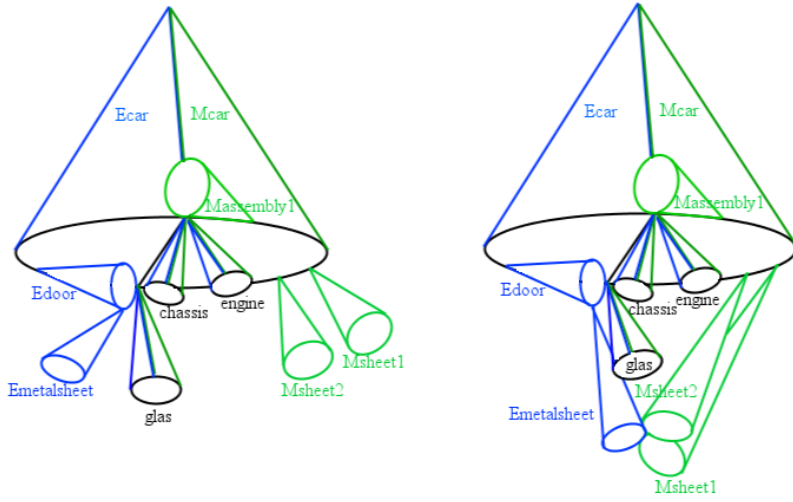


Figure 4 Graphical representation of the use case

5 Discussion

Unlike current static views (EBOM/MBOM), the proposed graphical representation integrates interface relationships by creating a model that physically evolves under design/manufacturing constraints. Figure 4 illustrates thus, based on this simple case study, an innovative ability of this model: dynamics. The addition of the property *is_linked_to* between **Emetalsheet**, **Msheet1** and **Msheet2** on the figure left side of the figure 4 has indeed strained the right side of this figure. Regarding the data extraction, in our study case, the information of the EBOM and MBOM are obtained manually and implemented in the ontology, since it is a very simple case. In more complex cases, the information about the items could be extracted automatically from the design and manufacturing information systems. However, we think that the implementation of the different properties can not be done automatically, as they result from a very complex and company specific workflow. The goal is rather to provide tools to make these

manual mappings as easy as possible to the user. In this paper, the graphical representation is obtained thanks to the ontology. However the opposite situation is possible, where the engineer is directly working on the graphic interface. By imposing geometrical constraints, he/she automatically instantiate individuals and properties in the ontological model. The mapping between the design and the manufacturing views -that have to be manual- could also be edited in a simple and visual way. This approach could be extended to other stages of the product lifecycle than design and manufacturing, providing thereby a global modelisation designed to ensure interoperability in PLM.

6 Conclusion

A new product model was introduced in order to make the Engineering and Manufacturing BOMs consistent. This model was so-called *minimal* since it was designed with the intent to enable the exchange of BOMs between PDM and ERP systems, while retaining a way to update both BOMs in case of an event occurring in both BOMs. This model uses an ontological representation including specific domain rules. A graphical representation of this model is presented, which is a 3d view of the items and their links, that contains all the information needed. Moreover, this model enables to express local and specific user views, as it contains the design, manufacturing and interface information. Further work will deal with the way to include a temporal (i.e. a process based) description.

References

- [1] CIMdata. PLM Market Growth in 2008. 2009.
- [2] T. Paviot. *A Methodology for solving interoperability problems in the field of Product Lifecycle Management*. PhD thesis, École Centrale Paris, France, 2010.
- [3] D. Alberts and R. Hayes. *Power to the edge : command, control in the information Age*. OTAN publication - CCRP, 2005.
- [4] P. Wegner. Interoperability. *ACM Computing Survey*, 28 (1):258–287, 1996.
- [5] F. Noel and L. Roucoules. The PPO design model with respect to digital enterprise technologies among product life cycle. *International Journal of Computer Integrated Manufacturing*, 21 (2):139–145, 2008.
- [6] EIF. European interoperability framework - White paper. pages 1–40, 2004.
- [7] A. Ben Kheder, S. Henry, and A. Bouras. An analysis of the interaction among design, industrialization and production. *Proceedings of 7th PLM Conference, Bremen, Germany*, 2010.
- [8] ISO-14258-1998. Systèmes d’automatisation industrielle - concepts et règles pour modèles d’entreprise. *International Standard Organization*, 1998.

- [9] A. Tursi, H. Panetto, G. Morel, and M. Dassisti. Ontological approach for product-centrics information system interoperability in networked manufacturing enterprises. *Annual Reviews in Control*, 33 (1):238–245, 2009.
- [10] T. Galeta, M. Klajin, and M. Karakasic. Product model suited for the ERP system. *9th International Design Conference-Design*, 2006.
- [11] T. Paviot, V. Cheutet, and S. Lamouri. A generic multiCAD/multiPDM interoperability framework. *International Journal of Services Operations and Informatics*, 6(1-2):124–137, 2011.
- [12] T. Paviot, V. Cheutet, and S. Lamouri. Federate design and logistics through Product Lifecycle Support standard. *6th Int. Conf. Product Lifecycle Management, Bath, United Kingdom*, 2009.
- [13] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43 (5):907–928, 1995.
- [14] S. Lee, H. Chae, C.H. Kim, and K. Kim. Design of product ontology architecture for collaborative enterprises. *Expert Systems with Applications*, 36 (2):2300–2309, 2009.
- [15] F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the semantic web. *Computer science*, 2605:228–248, 2005.
- [16] F. Baader, D.L. Mc Guinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logics Handbook*. Cambridge university press edition, 2003.
- [17] A. Matsokis and D. Kiritsis. An ontology-based approach for product lifecycle management. *Computers in industry*, 61:787–797, 2010.
- [18] X. Fiorentini, S. Rachuri, H. Suh, J. Lee, and R. Sriram. An analysis of description logic augmented with domain specific rules for the development of product models. *Journal of Computing and Information Science in Engineering*, 10:1–13, 2010.
- [19] J.C. Hernandez Mathias, H. Perez Garcia, J. Perez Garcia, and A. Vizan Idoipe. Automatic generation of a bill of materials based on attribute patterns with variant specifications in a customer-oriented environment. *Journal of Material processing Technology*, 199:431–436, 2008.
- [20] P.W. Stonebraker. Restructuring the bill of material for productivity: A strategic evaluation of product configuration. *Int. J. Production Economics*, 45:251–260, 1996.
- [21] J. Orlicky. Net change material requirements planning. *IBM Systems Journal*, 12 (1):2–29, 1973.
- [22] T. Paviot, V. Cheutet, and S. Lamouri. Semantic tags for generative multiview product beakdown. *7th Int. Conf. Product Lifecycle Management, Bremen, Germany*, 2010.
- [23] Protégé ontology editor. <http://protege.stanford.edu>. *Stanford University*.