A generic MultiCAD/MultiPDM interoperability framework

Thomas Paviot, Samir Lamouri, Vincent Cheutet

To cite this version:

Thomas Paviot, Samir Lamouri, Vincent Cheutet. A generic MultiCAD/MultiPDM interoperability framework. International Journal of Services Operations and Informatics, 2011, 6 (1/2), pp.124-137. 10.1504/IJSOI.2011.038322 . hal-00799837

HAL Id: hal-00799837
https://hal.science/hal-00799837

Submitted on 12 Mar 2013

A generic MultiCAD/MultiPDM interoperability framework

Thomas Paviot*, Samir Lamouri and Vincent Cheutet

Supméca/LISMMA,
3, rue Fernand Hainaut,
93407 Saint-Ouen, France
Fax : +33(1)149452991
E-mail: thomas.paviot@supmeca.fr
E-mail: vincent.cheutet@supmeca.fr
LCPI/Arts et Métiers ParisTech,
151, bd de l'Hôpital,
75013 Paris, France
Fax : +33(1)44246359
E-mail: samir.lamouri@ensam.eu
*Corresponding author

Abstract: Many Small and Medium Enterprises (SME) work as sub-contractors (or co-contractors) for several clients for the design of mechanical components. During the design process, they must use a variety of Computer Aided Design (CAD) softwares and connect all the Product Data Management (PDM) systems of their customers. After defining the specific needs of these companies, we show that the available commercial CAD/PDM integrations, as well as the current literature, are inappropriate for a multiCAD/multiPDM collaborative design. Are first defined the few simple processes required to ensure an efficient collaboration. Then, the instantiation of these processes in our CAD/PDM integration can be split into four points: the general definition of a CAD product structure tree and its associated model, the conversion algorithm of the product structure to an Engineering Bill Of Material (EBOM), the creation of an Unified Modeling Language (UML) data model, an implementation based upon Component Object Model (COM) and Service Oriented Architectures (SOA) technologies. We conclude by presenting the results obtained from the demonstrator developed.

Keywords: Computer Aided Design; Product Data Management; Interoperability; Engineering Bill Of Material; EBOM; Service Oriented Architectures; SOA.

Reference to this paper should be made as follows: Paviot, T., Lamouri, S. and Cheutet, V. (2010) 'A generic MultiCAD/MultiPDM interoperability framework', *Int. J. Services Operations and Informatics*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Dr. Thomas Paviot is graduated from the Arts et Métiers ParisTech engineering school and the École Normale Supérieure, Cachan, France. He received his PhD in engineering science in 2010 from the École Centrale Paris. His research area is focused on interoperability issues in the field of Product Lifecycle Management, from the business needs and workflows to the underlying IT architecture and implementation, and especially adresses the design and production stages.

Pr. Samir Lamouri is a Professor of Mechanical Engineering and Information Technology in the Department of Mechanical Engineering at Arts et Métiers ParisTech, France. He has been involved in several research projects about supply chain management and information systems. His research interests deal with supply chain management, information sharing and exchange, PLM, APS, ERP systems, collaborative practices, and their impacts on enterprise performance.

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. He received his PhD in mechanical engineering in 2006. His main research area is related to information flows between design and production based on IT systems like PDM or ERP.

1 Introduction

In the current context of concurrence increase, design-time and time-to-market reduce, the concept of extended enterprise, where heterogeneous enterprises are linked at one moment and for one specific project, has become a standard for new product projects, especially for the automotive and aeronautical industries. In this case, subcontractors may become co-partners of the product development, with in particular the charge of designing a specific sub-part of the system.

To achieve such organisation, each operator of the project should access to right design information, at the right time, in a comprehensible format for him. It implies to ensure a consistent digital framework, where heterogeneous systems have to communicate. This implies the use of a coherent information system to ensure project/product data coherency (Cui et al. (2006)). Among all the possible authoring tools and data management systems that can be used in the extended enterprise field, we will focus in the proposed approach on the mechanical Computer Aided Design (CAD) tools and Product Data Management systems (PDM).

For major projects, the chain of subcontracting may include several levels. Small and Medium Enterprises (SME) involved in these projects, acting as subcontractors at level 3 or 4, are located at the crossroads of several extended enterprises (several of these major groups can be their customers). Designers have so to use many CAD systems and connect the PDM systems of their customers, as well as their own PDM. Commercial CAD/PDM integrations are often expensive, and require significant training before they can be used. However, SME do not have the technical skills and financial resources to deploy these solutions quickly and efficiently. The CAD/PDM integration is then a major issue for these companies, since the ability to integrate the extended enterprises information technology system tends to become an order qualifier.

On one hand, SME were identified as having the following needs (Kadiri et al. (2009)):

- a low-cost CAD/PDM integration, easy to use, deploy and maintain,
- an installation that would not cause any important modification of the current system, both on server and client side,
- an extensible integration, i.e. easily allow the interoperability with another CAD or PDM system.

On the other hand, the following specific needs were expressed by designers:

- export the product structure to the PDM in order to manage an EBOM and CAD documents dependencies,
- allow an asynchronous collaborative design on a part of the mock up, enable data exchange with customers or suppliers,
- enable a project review based on 3D visualization.

As a consequence, two main issues appear crucial for SME: reduction of design time and risks of mistakes in the product structure management thus enabling them to compete for a call to tender.

To achieve these two issues, we propose a multiCAD/multiPDM integration framework, based on two main concepts : first a mediation architecture between the different involved systems, allowing a great agility in terms of system connection, and second, data exchanges between these systems based on standards format, like Standard for the Exchange of Product model data (STEP).

The article is structured as follow. Section 2 overviews different approaches existing in the literature on the CAD/PDM interoperability issue. Section 3 demonstrates the validity of our approach according to a graph-based model of dependencies in the Digital Mock-Up (DMU) description, and presents the conversion rules of our MultiCAD/MultiPDM integration framework to export the product structure to an Engineering Bill Of Materials (EBOM). Section 4 describes the implementation architecture and the results obtained on different use cases. Finally, section 5 presents the conclusions and open new perspectives.

2 CAD/PDM interoperability issue

The problem of CAD/PDM integration can be seen as a problem of system interoperability as seen before. Among the exhaustive list of possible definitions for interoperability that has been listed by Baina (2006), we used the proposition of EIF (2004) for whom interoperability can be reached at different levels:

- Technical layer (exchanged data et messages) : to exchange information, we should first have a transport vector available between the two systems that communicate, i.e. a functional data vector,
- Semantic layer (information and services sharing): a functional data vector is not enough, if the exchanged data cannot be understood by the two systems. Data, with attached meaning, become so information that can be treated by the two systems,
- Organisational layer (interactions between business unit/process/people through the organisation): an adapted organisation should be thought to ensure the information exchange.

These layers have to been simultaneously faced to guarantee a complete interoperability between information systems. In this article, we will only tackle the first two layers (i.e. technical and semantic) in a first time, the organisation layer implying problems relative to processes and their modelling that we will not present here.

2.1 Technical interoperability

In this section we will study and compare the possible architecture allowing to realise the technical interoperability between several CAD and PDM systems. To guarantee the information exchange between systems, each couple $(i; j)$ of systems should be interoperable, i.e. at least one path linking each couple of node $(i; j)$ should exist in the chosen network topology. Among all the possible network topologies, two types of architectures arise from the study of the PLM interoperability literature (Guyot et al. (2007)):

- A point-to-point architecture, whose topology is a complete graph, in which each system i is connected to the system j through a specific interface;
- A mediator architecture (”star” topology) in which a new system is added in the middle. In this architecture, first introduced by Wiederhold (1992), the added system is called information mediator.

If we compare the two architectures either in terms of Total Cost of Ownership (TCO), or in terms of system agility, the mediator architecture is the most-suited for the system interoperability issue, especially in the context of SME integration in the extended enterprise.

Since the article of Wiederhold (1992), SOA (Service Oriented Architecture) (Gottschalk (2000)) and Web Services (Booth et al. (2004)) have allowed the development of the concept of service-oriented mediator. First presented by Benaben et al. (2008), this service-oriented architecture has been demonstrated as the most-relevant one for information systems interoperability in a larger context by Paviot et al. (2009).

2.2 Semantic interoperability

The standard ISO-14258-1998 (1998), relative to the enterprise modelling, precise that the semantic interoperability between two (or more) information systems of enterprise can be tackled in three different ways:

- Integration: a common standard of data model is used for all the system components. The integration process implies so to merge the data models;
- Unification: a meta-model common to all system components furnishes a way to establish semantic associations;
- Federation: distinct models are dynamically associated. This approach is usually using semi-automatic tools, based on heuristic methods that mainly compare the terminology and the data structure to detect couples of concepts that are linked at the semantic level (similarity or equivalence) and called mapping.

According to the discussion of Hoffmann (2008) on these three different approaches, the unification approach has been chosen in our proposition. In such case, the study of the literature on the PLM interoperability enlightens two different strategies: authors can either define an *ad hoc* data model or use and implement a standard data model.

A priori the solution of a neutral format translation appears to be adapted (Fenves et al. (2005)). Among the different existing standards existing in the PLM field, STEP file format is an internationalized standard (Pratt (2005)) that offers various means of storing, exchanging and archiving the product data in a long-term approach. Several authors so used this standard to implement a CAD/PDM interoperability.

Oh et al. (2008) develop a CAD/PDM integration based upon EXPRESS mapping language and an UML mapping diagram. Methodology of Zhang et al. (2000) maps IGES, STEP AP203 and STEP AP209 standard. However, although the STEP Schema has been specified for exchanging data usually stored by PDM systems (Machner et al. (1998)), the use of STEP file format to extract the DMU information leads to two problems: CAD editors partially implement the ISO standard (Oh et al. (2008)) and STEP processor acts like a filter on native data that is problematic in the case of an homogeneous CAD/PDM integration, *i.e.* both designers working with the same CAD tool. Song et al. (2007) specifies a CAx/PDM integration platform based on product model defined by STEP AP203. The results of such works have been implemented on the basis of Cooperative Design Modeller (CoDeMo) core.

In order to validate the two aspects of the interoperability framework we propose, we study in the next section a graph description of both CAD and PDM data models, to enlighten the similarities and the differences between the two models.

3 Proposed approach

The concept of *item* can be considered as the key-item of the enterprise since it is used by every department: design, maintenance, manufacture, distribution, inventory management and Material Requirement Planning (MRP). Each of them view the BOM 'as maintained', 'as built', 'as designed' etc. Parts are necessary to manage the engineering changes requests as well as the product configuration (Estublier et al. (2007)). In order to allow the connection to the

Information Technology (IT) system of the extended enterprise, we consider that our CAD/PDM integration should focus on EBOM checking, reading and updates.

On the contrary of the manufacturing field where the notion of Manufacturing Bill Of Material (MBOM) is perfectly defined and shared since the definition of Orlicky (1974), there is not a univocal definition or proposition of product structuring in the design process: 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)). In our approach, we consider that the possible CAD/PDM integration exist in a specific extended enterprise, and so the product breakdown process and its attached semantics is shared between all the actors of the product development process, even if they do not belong to the same organization.

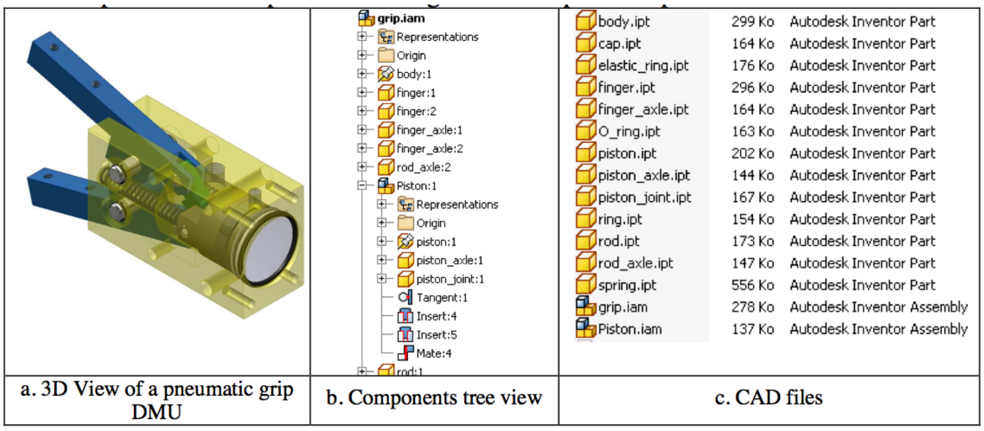
We then have two different structures describing the product: a CAD document structure, which defines the links between CAD files, and an EBOM that describes the structural links between PDM parts. This section presents the methodology used to convert a CAD product structure and all the dependencies to this double structure in the PDM system.

3.1 Dependencies in a DMU

Although each CAD software implements its own data model and graphical user interface, we noticed that the ones we studied share a set of common basic features, as well as the same way to store data on the hard disk. In particular, we noticed that a DMU is completely defined by:

- a set of CAD files that describe the geometry of the system. Each software uses a proprietary file format to store geometric data, but the information is structured by two kinds of files: parts and assemblies (CATPart and CATProduct for CATIA V5^{TM}, SLDPRT and SLDASM for SolidWorks^{TM}, IPT and IAM for Inventor^{TM}, PRT and ASM for Pro/E^{TM}),
- a set of files that define the different configurations of a part. These are usually text CSV formatted files that can come from a spreadsheet application,
- a set of components linked with parent/child relations. Components can be instances of CAD files.

These dependencies are presented for a simple example: fig.1.a presents the 3D view of a pneumatic grip, that is defined in Autodesk Inventor^{TM} CAD Software by the tree view available in the CAD environment (fig.1.b) and the list of CAD files necessary for the DMU (fig.1.c)



```
grip.iam
  Representations
  Origin
  body:1
  finger:1
  finger:2
  finger_axle:1
  finger_axle:2
  rod_axle:2
  Piston:1
    Representations
    Origin
    piston:1
    piston_axle:1
    piston_joint:1
    Tangent:1
    Insert:4
    Insert:5
    Mate:4
  rod:1
```

body.ipt	299 Ko	Autodesk Inventor Part
cap.ipt	164 Ko	Autodesk Inventor Part
elastic_ring.ipt	176 Ko	Autodesk Inventor Part
finger.ipt	296 Ko	Autodesk Inventor Part
finger_axle.ipt	164 Ko	Autodesk Inventor Part
O_ring.ipt	163 Ko	Autodesk Inventor Part
piston.ipt	202 Ko	Autodesk Inventor Part
piston_axle.ipt	144 Ko	Autodesk Inventor Part
piston_joint.ipt	167 Ko	Autodesk Inventor Part
ring.ipt	154 Ko	Autodesk Inventor Part
rod.ipt	173 Ko	Autodesk Inventor Part
rod_axle.ipt	147 Ko	Autodesk Inventor Part
spring.ipt	556 Ko	Autodesk Inventor Part
grip.iam	278 Ko	Autodesk Inventor Assembly
Piston.iam	137 Ko	Autodesk Inventor Assembly

a. 3D View of a pneumatic grip DMU

b. Components tree view

c. CAD files

Figure 1 Dependencies for a simple DMU (screenshots from Autodesk Inventor^{TM})

These data must be stored into the PDM to ensure their coherence during the design process.

As a consequence, we define an directed graph-model of these dependencies that leads to the data model to be implemented (fig.2).

Three different types of elements (i.e. the vertices of the proposed graph model) have been defined:

- F is the set of all the necessary and sufficient files needed to completely describe the geometry of the digital mock-up. Each of these files describes a part of the system.

- I is the set of components that appear in the design tree view that is generally displayed on the left part of the CAD software window. These components can be, or not, an instance of an element of F.

- C is the set of files that define different configurations for a part or an assembly.

Each file $f \in F$ can be instantiated more than once, i.e. the same CAD file can be instantiated several times in the assembly. In the example of fig.1, the assembly is composed by 2 fingers (present in the tree view), but only one finger file exists. The current trend to standardize products leads to a reduction in the elements of F to describe the geometry of a product. We can then write the following cardinality property:

$$0 \leq card(C) \leq card(F) \leq card(I) \tag{1}$$

Four different types of directed links (i.e. the arcs of the proposed graph model) between these elements can be identified:

- structural links (S_L) between elements of I,

- geometrical description links (G_L) between an element of I and element of F,

- configuration links (C_L) between elements of F and elements of C,
- structural saved links (S'_L) between elements of F. They can come from a S_L structural saved link or parametric link between two parts ("in-context" design, parameters relationship).

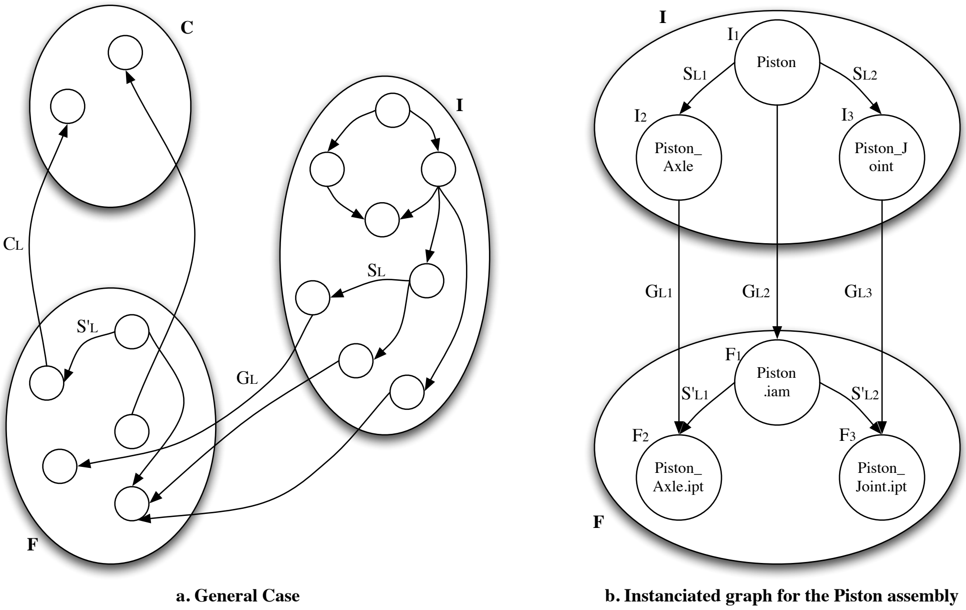


Figure 2 Graph model for the instances/files dependencies

Both (S_L) and (S'_L) are necessary to have complete information about dependencies: (S_L) provides the information about the number of instances whereas (S'_L) provides the information of which files are necessary to open the DMU in the CAD software. By the same reasoning as for the vertices, we can establish the following cardinality property:

$$0 \leq card(S') \leq card(S) \tag{2}$$

The set of geometrical description links (G_L) can be seen as a surjective function from I to F, which is in adequation with the cardinality property between the two sets.

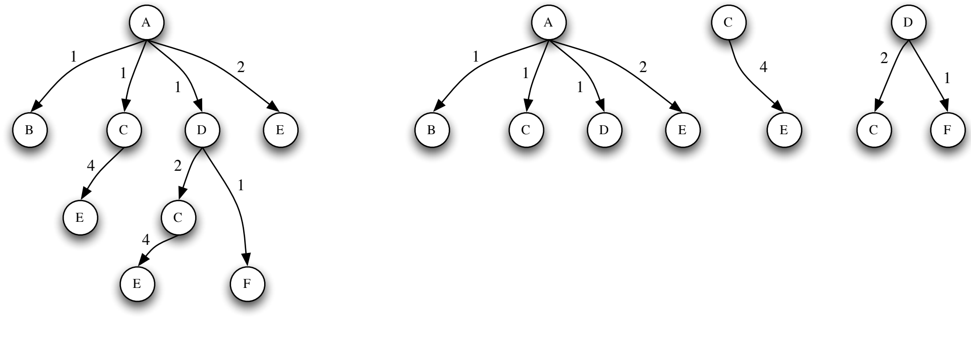
Let D be the set including all DMU data ($D = C \cup F \cup I$) and L the family of dependencies between these data ($L = S_L \cup G_L \cup C_L \cup S'_L$). The graph $DMU(D, L)$ contains the required information that has to be stored in the PDM database. We define the subgraph $PS(I, S_L)$ as the *digital product structure.*

Since we defined the graph $DMU(D, L)$ as the model of the necessary dependencies to keep the CAD data coherent, once the current work of the designer is saved to his local hard disk, DMU is a connected and directed graph. An element of I cannot handle a self-reference so there cannot be any loop in $PS(I, SL)$ (3).

$$\text{if} \quad \exists (I_i, I_j) \in S_L \Rightarrow (I_j, I_i) \notin S_L \tag{3}$$

3.2 as-designed EBOM graph model

Similarly to the MBOM (Orlicky (1974)), we can define an EBOM as a graph $E(P, L)$, the vertices P being parts and the arcs L structural links (cf. fig.3). E is a directed, weighted 1-graph. Weight w_i of arc L_i is given by the number of elements of its parent. The EBOM can be exploded into n 1-level EBOM in order to simplify the treatment.



a. General EBOM

b. 1-level exploded EBOM

Figure 3 EBOM graph model

3.3 Product Structure to EBOM conversion rule

The two structures defined in the previous sections, and especially $PS(I, S_L)$ and $E(P, L)$, are quite similar in terms of graph property. Nevertheless, the semantics between them may differs and we then have to determine which CAD components have to be converted to PDM parts and the conversion rules that need implementing. The conversion rules must solve these two issues:

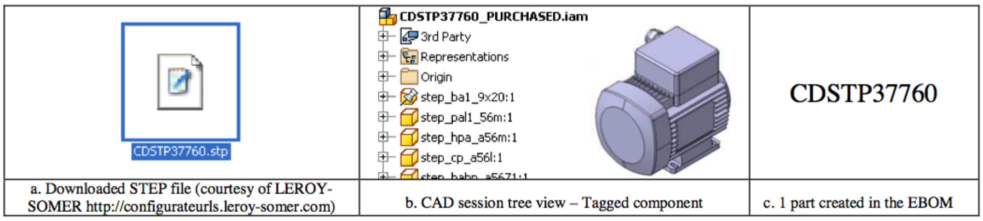
- the rule for converting CAD instances into parts
- the rule of constructing EBOM links from the product structure, i.e. define an application from $PS(I, S_L)$ to $E(P, L)$.

The first approach we investigated is the one that is usually implemented in the CAD softwares: each new CAD component, i.e. a vertex of the $PS(I, S_L)$ graph, corresponds to a PDM part, i.e. a vertex of the $E(P, L)$ graph. But, by using this rule, three issues have to be faced.

The first issue correspond to the core difference between the arcs of the $E(P, L)$ graph, that are weighted, and the ones of the $PS(I, S_L)$ graph, that are not. For instance, in the example of the grip (fig.1), two fingers exist in the digital product structure, whereas the EBOM is composed by one item “finger” linked to the item “grip” by an arc with a weight equal to 2. In this case, we have to check first the $DMU(D, L)$ graph: the two vertices “finger” present in I are linked to the same vertex “finger.ipt” in F, i.e. the two instances of the digital product structure are attached to the same geometry and so can be grouped together in the EBOM. To obtain such result, we use the following algorithm: $\forall I_i \in I$, a recursive parsing is performed to get all the child components $I_{k,l,m...}$. For each of these $(I_i I_{k,l,m...})$ found links, the corresponding EBOM parts are queried then the link (PP) is created with $w = 1$ otherwise w is incremented if the link is already created. The algorithm is initialized with the root node I_0 and recursion ends when I_i does not have any child.

The second issue concerns the instances referring to a part with no mass: in an early design stage, each part may be modeled by few geometric characteristics (point, line, plane, etc.) on which future mass will stand. Another usual design methodology uses master sketches that drive the geometry of the DMU, since they allow a robust and flexible design. These master sketches must not be converted to an EBOM part: they have to be considered as tools in the DMU building process. We decide to put an *MS* tag (for Master Sketch) in the name of the instance, to specify to the translator that it should be skipped. These considerations lead to the following single rule: all instances that are not “MS” tagged are converted into parts.

The third issue appears when the designer uses a standard component or a purchased component. As we can see in fig.4, he uses the 3D model of the component obtained from the supplier database (cf. fig.4.a). This model is generally available in the STEP format, and once it is instantiated in the CAD session, many components are created (cf. fig.4.b). In this case, it is not necessary to create the PDM parts corresponding to each item, but only the one corresponding to the root node of the structural tree (cf. fig.4.c). This is achieved by adding a *PURCHASED* tag to the instance name.



a. Downloaded STEP file (courtesy of LEROY-SOMER http://configurateurls.leroy-somer.com)

b. CAD session tree view – Tagged component

c. 1 part created in the EBOM

Figure 4 Product structure conversion for standard components

These three issues can be considered as operation of type vertices clustering on the graphs, and so do not change the graph structure. As a consequence, the transformation from $PS(I, S_L)$ to $E(P, L)$ is a surjective application, which validates our approach.

4 Implementation

This section explains the implementation of the previous algorithm: subsection 4.1 focuses on the data model whereas 4.2 presents the functional architecture of the implementation.

4.1 Data model

Eynard et al. (2006) define a complete UML class diagram of a product structure. However two problems subside: on one hand, the dependencies between the CAD files are not described; on the other hand this model is strongly linked to the PDM system they study and is not easily portable to another system. We propose the below model (fig.5) that uses elementary classes available on any PDM system (Part, CAD Document, Generic Document) and for which implementation does not require any customization on the PDM server side. The double Part and CAD document structures appear on this diagram. The "Structure links" connect the "Part Master" class with the "Part" class that can be iterated during the design process. Each part is described by 3 "Generic Documents" that embed the 3D Preview, exchange files and configuration table files. Each "Part" is described by 0 or 1 "CAD Document" which refers to a native CAD file.

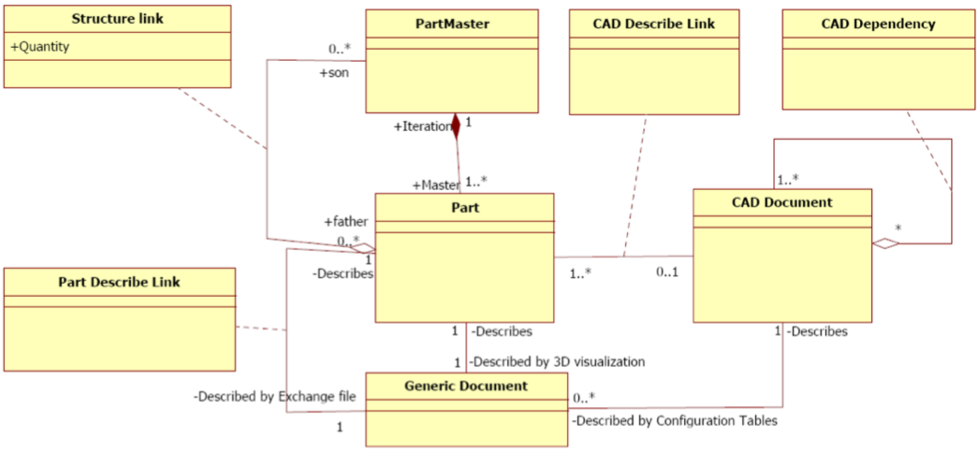


Figure 5 UML data model of the converter

4.2 Implementation architecture

The implementation architecture relies on two technologies (cf. fig.6):

- Component Oriented Model (COM): this Microsoft^{TM} Windows^{TM} specific technology is designed to allow the automation of manual processes. The data model of each CAD software is specific, but we implemented a COM client that relies on a set of generic basic classes/methods that are available in each software. A simple XML mapping file is used to instantiate our generic client object. COM client parses product structure, export files as STEP or 3DXML.

- Web Services (Service Oriented Architecture - SOA): a service is an implementation independent interface with an explicit definition; a service is loosely coupled, location-transparent and called by interoperable communication protocol; a service encapsulates re-usable business functions.

During the last years, PDM editors have made significant efforts to provide standardized SOA (Lee et al. (2007)) that make their system interoperable. Simple Object Application Protocol (SOAP) (WWW Consortium (2007)) requests deal with part/documents/links creation on the PDM.

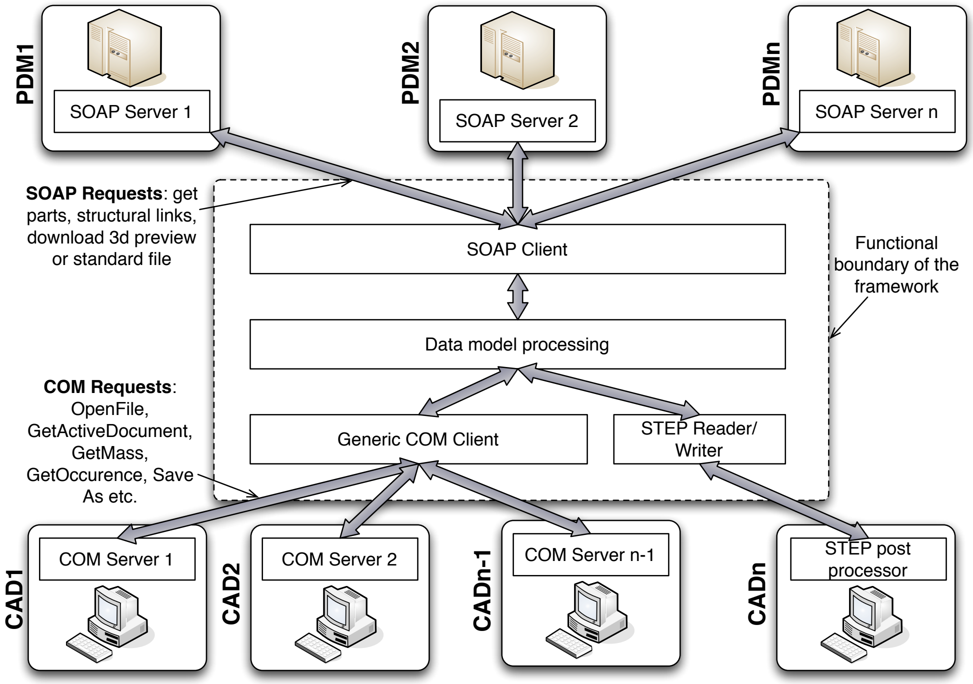


Figure 6 Architecture implementation based on COM and SOAP technologies

For the CAD softwares that do not provide any COM server, or that run on other operating systems, we developed a simple STEP parser that can extract the product structure information (Oh et al. (2008)).

4.3 3D Visualization

To achieve the 3D visualization of our integration, a number of file formats are available that can satisfy the needs of all users by allowing 3D data, and sometimes other data such as 2D drawings and documents, to be viewed from a fairly simple and free of charge viewing tool. These include formats such as UGS'JT, Dassault Systèmes'3DXML, 3D Industry Forum's (3DIF) U3D, Tech Soft's HOOPS Stream, and Web 3D Consortium's VRML and X3D.

3D visualization files are created from the CAD session with the appropriate COM method, exported to the PDM and converted into 3D preview "Generic Document" (cf. fig.5). These documents can be accessed by any authorized user even if the CAD software is not installed on his computer. It is especially useful for

PDM systems accessible from a lightweight client: a collaborative project review can be done from almost any computer, as long as the appropriate viewer is installed and configured.

4.4 Results

We developed a demonstrator that implements previous work. We identified four elementary processes that we implemented to achieve a simple CAD centric collaborative design:

- export of the CAD data to the PDM,
- download of the CAD data from the PDM,
- request a lockup of a part (checkout) and begin a new design process,
- upload modifications to the PDM (check-in) and allow other designers to access new part iteration.

The product structure of the example of fig.1 from Autodesk Inventor^{TM} was exported to PTC Windchill^{TM} PDM. We also investigated further with bigger and more complex DMUs (cf table.1).

DMU name	*Number of I*	*Number of parts*	*DMU file size*	*Total transfer time*
Grip	10	6	3.5Mb	5mn
Aircraft	100	90	200Mb	45mn

Table 1 Results of CAD export to a remote on-demand PLM platform

5 Conclusion and further works

In this paper, we presented a methodology to convert a CAD product structure to an EBOM, independently of the softwares used. We defined an UML data model that was implemented to provide an easy to use/deploy/maintain and low-cost bidirectional CAD/PDM bridge. Basic features of collaborative design work are fully functional but we still encounter problems to properly handle events sent by the CAD or the PDM when a modification occurs on either CAD or PDM sides.

Further short term work will focus on issues of change management in product structure and synchronization of data. An other issue is to develop a methodology to convert an Engineering Bill Of Material into a Manufacturing Bill Of Material (MBOM), so that our demonstrator can connect to an Enterprise Resource Planning (ERP) system (CIMdata (2006)) and cover a larger scope of extended enterprise needs (Paviot et al. (2009)).

References

Baïna, S. (2006) 'Interopérabilité dirigée par les modèles : Une Approche Orientée Produit pour l'interopérabilité des systèmes d'entreprise', *Thèse de doctorat de l'Université Henri Poincaré, Nancy I.*

Bénaben F., Touzi J., Rajsiri V., Truptil S., Lorré J., Pingaud H. (2008) 'Mediation Information System Design in a Collaborative SOA Context through a MDD Approach', *MDISIS Workshop, CAISE 08 conference*, Montpellier.

Booth D., Haas H., McCabe F., Newcomer E., Champion M., Ferris C., Orhcard D. (2004) 'Web Services Architecture', *W3C Working Group Note*, 11/2/2004.

CIMdata (2006) 'PLM and ERP Integration: Business Efficiency and Value', http://www.cimdata.com.

Cui J., Qi G. (2006) 'Research on Integration Technology for Product Lifecycle Management System', *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA06).*

EIF (2004) 'European Interoperability Framework - White Paper', pp.1–40.

Estublier J., Vega G. (2007) 'Reconciling Software Configuration Management and Product Data Management', *ESEC/FCE07.*

Eynard B., Gallet T., Roucoules L., Ducellier G. (2006) 'PDM system implementation based on UML', *Mathematics and Computers in Simulation*, Vol.70, Issues 5-6, pp.330–342.

Fenves S.J., Sriram R.D., Subrahmanian E., Rachuri S. (2005) 'Product Information Exchange : Practices and Standards', *Journal of Computing and Information Science in Engineering*, Vol.5, pp.238–246.

Fowler J. (1995) 'STEP for Data Management, Exchange and Sharing, Technology Appraisals', ISBN 1-871802-36-9.

Gottschalk K. (2000) 'Web services architecture overview', *IBM white paper*, IBM developer Works , 1er septembre 2000.

Guyot E., Ducellier G., Eynard B., Girard P., Gallet T. (2007) 'Product data and digital mock-up exchange based on PLM', *Proceedings of Conference on Product Lifecycle Management (PLM07)*, pp.243–252.

Hoffmann P. (2008) 'Similarité sémantique inter-ontologies basée sur le contexte', *Thèse de doctorat de l'Université Claude Bernard - Lyon 1.*

ISO-14258-1998 (1998) 'Systèmes d'automatisation industrielle - Concepts et règles pour modèles d'entreprise', International Standard Organization.

Kadiri S.E., Pernelle P., Delattre M., Bouras A. (2009) 'Current situation of PLM systems in SME/SMI : Surveys results and analysis', *6th International Conference on Product Lifecycle Management*, Bath - GB.

Lee T., Lim J., Shin J., Myung S., Choi M., Baek S., Kim J., Oh J., Lee D., Han Y. (2007) 'An implementation methodology of SOA based PLM system', *Proceedings of International Conference on Product Lifecycle Management (PLM07)*, pp.303–310.

Machner B., Ungerer M. (1998) 'Mapping the user requirements from the VDAORG with the STEP PDM Schema', *Product Data Journal*, Vol.5.

Oh Y., Han S., Suh H. (2001) 'Mapping product structures between CAD and PDM systems using UML', *Computer-Aided Design*, Vol. 33, Issue 7, pp.521–529.

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

Paviot T., Cheutet C., Lamouri S. (2009) 'Design and logistics IT federation through Product Lifecycle Support standard', *Proceedings of International Conference on Product Lifecycle Management (PLM09).*

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

Song H., Eynard B., Roucoules L., Lafon P., Charles S. (2007) 'Beyond geometric CAD system : implementation of STEP translator for multiple-views product modeller', *International Journal of Product Lifecycle Management*, Vol. 2, Issue 1, pp.1–17.

Tomovic C., Walton A., Ncube L., Grieves M., Birtles B., 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.

Wiederhold G. (1992) 'Mediation in the architecture of future information systems', *IEEE computer*, Vol. 25, Issue 3, pp.38–49.

World Wide Web Consortium (2007) 'Simple Object Access Protocol', http://www.w3.org/2000/xp/Group.

Zhang Y., Zhang C., Wang H.P. (2000) 'An Internet based STEP data exchange framework for virtual enterprises', *Computers in Industry*, Vol. 41, Issue 1, pp.51–63.