



HAL
open science

MultiCAD/MultiPDM integration framework

Thomas Paviot, Pascal Morenton, Vincent Cheutet, Samir Lamouri

► **To cite this version:**

Thomas Paviot, Pascal Morenton, Vincent Cheutet, Samir Lamouri. MultiCAD/MultiPDM integration framework. PLM08, IFIP WG5.1, Jul 2008, Seoul, South Korea. pp 101–110. hal-00709562

HAL Id: hal-00709562

<https://hal.science/hal-00709562>

Submitted on 19 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.

MultiCAD/MultiPDM integration framework

Thomas Paviot

LISMMA - Supméca

3, rue Fernand Hainaut – 93047 Saint-Ouen – France

+33(0) 1.49.45.29.38

thomas.paviot@supmeca.fr

Pascal Morenton

Laboratoire Génie Industriel – Ecole Centrale Paris – France

Grande Voie des Vignes – 92295 Châtenay-Malabry Cedex

+33(0) 1.41.13.14.52

pascal.morenton@ecp.fr

Vincent Cheutet, Samir Lamouri

LISMMA - Supméca

3, rue Fernand Hainaut – 93047 Saint-Ouen – France

+33(0) 1.49.45.29.38

vincent.cheutet@supmeca.fr ; samir.lamouri@supmeca.fr

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 CAD softwares and connect all the 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 UML data model, an implementation based upon COM and Service Oriented Architecture (SOA) technologies. We conclude by presenting the results obtained from the demonstrator developed.

Keywords: CAD/PDM integration, Product Structure, EBOM, SOA, UML.

1 Introduction

In order to cope with the necessity of reducing design time and time-to-market, large industrial companies (especially automotive or aeronautical companies) have become extended enterprises in which subcontractors become co-partners of the product development. The use of a coherent information system ensures project/product data coherency [1]. Each operator of the project accesses a single repository thus ensuring an effective collaborative work. In this paper, we focus 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. 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 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. 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, the SME were identified as having the following needs:

- 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.

Two issues appear as 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.

When working with CAD and PDM systems without their native integration (whether it is too costly or the editors did not develop it), designers must reintegrate the data manually in the PDM thus causing a waste of time and a big risk of error. Data exchange using a standard data format (like the Standard for the Exchange of Product model data (STEP) for example) is also possible, but the incomplete implementation of STEP by CAD editors limits the scope of features needed by users (these points are developed in chapter 3).

The main issues thus identified are: Is there a way to overcome limitations in the implementation of STEP standard to allow an efficient collaborative design? What is the process of conversion of a CAD product structure to an Engineering Bill Of Materials (EBOM)? After a brief description of the state-of-the-art in CAD/PDM integration (chapter 2), we propose an approach based on a technique that obtains the information from the product structure before the STEP processor. We then develop a graph-based model of dependencies in the Digital Mock-Up (DMU) description, and conversion rules

MultiCAD/MultiPDM integration framework

to export the product structure to EBOM (cf. chapter 3). Chapter 4 is relevant to the implementation architecture and the results obtained.

2 CAD / PDM integration

In order to integrate a CAD system with a PDM system, a data translator has to be developed. According to Fowler [2], there are four different solutions for the data exchange problem: the manual reintegration of data, the direct translation, the neutral format translation and the product database sharing. The last two solutions are reasonable, flexible and adaptable. Direct translation is the most accurate solution, but the number of translators increases exponentially with the number of systems involved. Guyot *et al.* [3] calculate that for n systems, the number of translators is n^2-n for the direct translation, and only $2n$ with a neutral format. If we consider using a commercial native CAD/PDM integration, which is an example of direct translation, the total cost of ownership (TCO) is then $O(n^2-n)$. The exponential growth of TCO is not compatible with SME resources.

A priori the solution of a neutral format translation appears to be adapted [4]. STEP file format is an internationalized standard [5] that offers various means of storing, exchanging and archiving the product data in a long-term approach. Oh *et al.* [6] develop a CAD/PDM integration based upon EXPRESS mapping language and an UML mapping diagram. Methodology of Zhang *et al.* [7] maps IGES, STEP AP203 and STEP AP209 standard. However, although the STEP Schema has been specified for exchanging data usually stored by PDM systems [8], the use of STEP file format to extract the DMU information leads to two problems: CAD editors partially implement the ISO standard [6] 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 (cf. Chapter 3).

3 Proposed approach

The article is 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 [9]. In order to allow the connection to the Information Technology (IT) system of the extended enterprise, we consider that it is important that our CAD/DPM integration generates an EBOM. 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 chapter 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

CAD editors now offer the same kind of features. By studying the different CAD software that we are used to working with, we noticed that a DMU is completely defined by:

Paviot T., Morenton P., Cheutet V. and Lamouri S.

- 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™, SLDPRT and SLDASM for SolidWorks™, IPT and IAM for Inventor™, PRT and ASM for Pro/E),
- 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 on fig.1 for a simple example:

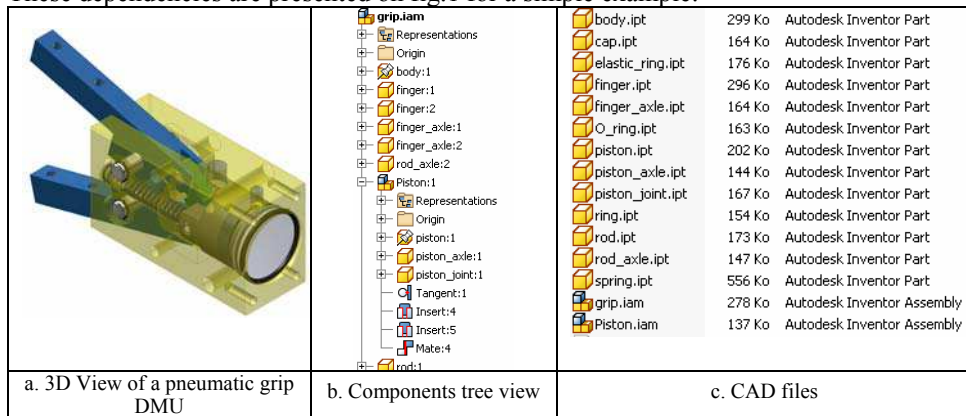


Figure 1 Dependencies for a simple DMU (screenshots from Autodesk Inventor™)

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

3.1.1 Graph model of dependencies

In this subchapter, we define a graph-model of these dependencies that leads to the data model to be implemented.

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.

We identified four different types of links between these elements:

- 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 structural saved links or parametric links between two parts (“in-context” design, parameters relationship).

MultiCAD/MultiPDM integration framework

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.

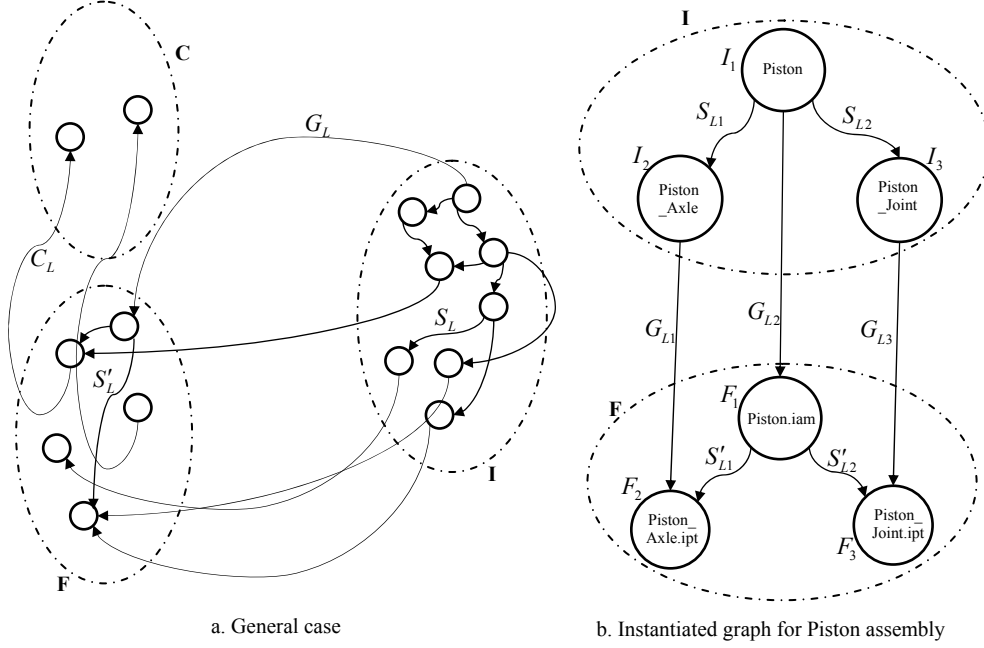


Figure 2 Graph model for the instances/files dependencies

Let D be the set including all DMU data ($D = C \cup F \cup I$) and L the family of dependencies between these data. 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 “product structure”.

3.1.2 Properties of this model

Each file in F can be instantiated more than once. 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:

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

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 reference to himself so if $\exists (I_i, I_j) \in PS \rightarrow (I_j, I_i) \notin PS$. There can not be any loop in $PS(I, S_L)$.

3.1.3 Product structure to export to PDM

Our integration must take into account the EBOM of the product. This approach is necessary to connect the Material Requirement Planning (MRP) system, or to enable the product configuration management.

3.2 “As Design” EBOM graph model

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

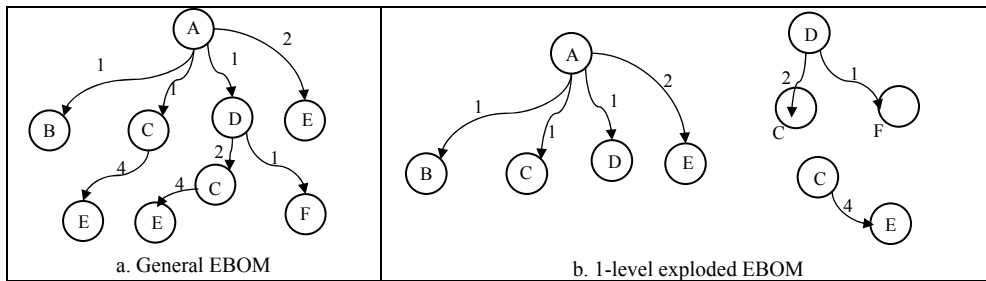


Figure 3 EBOM graph model

3.3 Product Structure to EBOM conversion rule

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 a morphism from $PS(I, S_L)$ to $E(P, L)$.

3.3.1 Converting CAD components to PDM parts

The first approach we investigated is the one that is usually implemented in the CAD softwares: each new instance of the $PS(I, S_L)$ graph corresponds to a part. This bijective application defines an isomorphism between $PS(I, S_L)$ and $E(P, L)$. A special case concerns the instances referring to a part with no mass: in an early design stage, each part may be modeled by a 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.

MultiCAD/MultiPDM integration framework

We must also take into account another aspect: when the designer uses a standard component or a purchased component (cf. fig.4), he uses the 3D model of the component obtained from the supplier database (cf. fig.4a). This model is generally available in the STEP format, and once it is instantiated in the CAD session, many components are created (cf. fig.4b). 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.4c). This is achieved by adding a “PURCHASED” tag to the instance name.

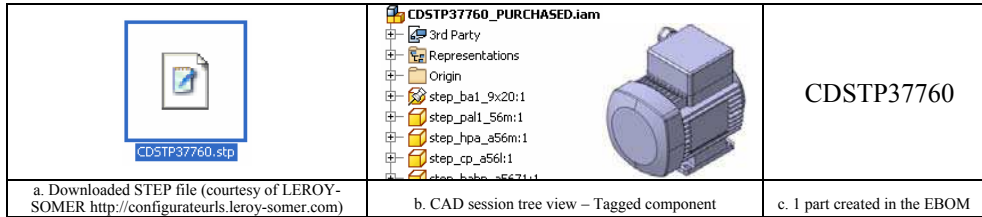


Figure 4 Product structure conversion for standard components

3.3.2 Converting CAD structural links to EBOM links

This step aims to build EBOM L_i and w_i as defined in 3.2. We use the following algorithm: for each I_i of I , a recursive parse 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 $(P_i P_{k,l,m...})$ is created with $w_i = 1$ or w_i 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.

4 Implementation

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

4.1 Data model

Eynard *et al.* [10] 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.

Paviot T., Morenton P., Cheutet V. and Lamouri S.

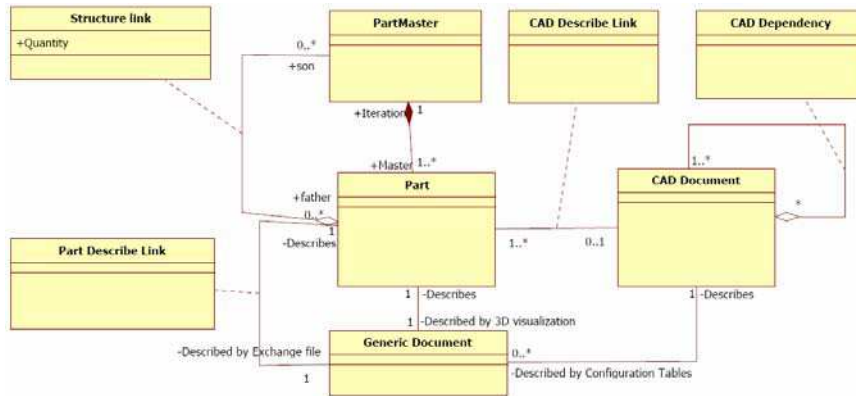


Figure 5 UML data model of the translator

4.2 Implementation architecture

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

- Component Oriented Model (COM): this Microsoft™ Windows™ 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): 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 many efforts to provide standardized SOA [11] that make their system interoperable. Simple Object Application Protocol (SOAP) [12] requests deal with part/documents/links creation on the PDM.

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 [6].

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's especially useful for PDM systems

MultiCAD/MultiPDM integration framework

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.

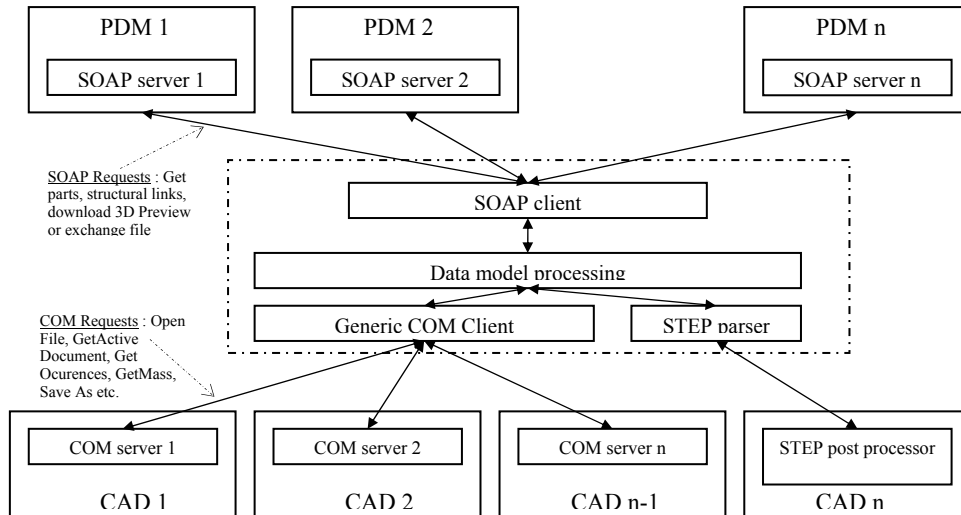


Figure 6 Implementation architecture based on COM and SOAP technologies

4.4 Results

We developed a demonstrator (released under the terms of a free software license) that implements previous work: CAD Shuttle [13]. 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.

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

DMU name	Number of I	Number of parts	DMU Size	Total export time
Grip	10	6	3.5Mb	5mn
Aircraft	100	90	200Mb	45mn

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

CAD Shuttle has been successfully tested with 2 other CAD software and 1 PDM system, using the same and simple graphical user interface.

5 Conclusion

In this paper, we presented a methodology to convert a CAD product structure to an EBOM. 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 both CAD or PDM sides. Further short term work will focus on issues of change management in product structure and synchronization of data. In the long term, we plan 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 [14] and cover a larger scope of extended enterprise needs.

References

- 1 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 (ISDA'06)*.
- 2 Fowler J. (1995) 'STEP for Data Management, Exchange and Sharing, Technology Appraisals', ISBN 1-871802-36-9.
- 3 Guyot E., Ducellier G., Eynard B., Girard P., Gallet T. (2007) 'Product data and digital mock-up exchange based on PLM', *Proceedings of PLM07*, pp.243--252.
- 4 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.
- 5 Pratt M.J. (2005), 'ISO 10303: the STEP standard for product data exchange, and its capabilities', *International Journal of Product Lifecycle Management*, 1, pp. 86--94.
- 6 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.
- 7 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.
- 8 Machner B., Ungerer M. (1998) 'Mapping the user requirements from the VDAORG with the STEP PDM Schema', *Product Data Journal*, 5.
- 9 Estublier J., Vega G. (2007) 'Reconciling Software Configuration Management and Product Data Management', *ESEC/FCE07*.
- 10 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.
- 11 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 PLM07*, pp.303--310.
- 12 World Wide Web Consortium (2007) 'Simple Object Access Protocol' – <http://www.w3.org/2000/xml/Group>
- 13 CAD Shuttle. <https://gna.org/projects/cadshuttle>.
- 14 CIMdata (2006), 'PLM and ERP Integration: Business Efficiency and Value', <http://www.cimdata.com>.