HAL
open science

A PLCS framework for PDM / ERP interoperability

Thomas Paviot, Vincent Cheutet, Samir Lamouri

To cite this version:

Thomas Paviot, Vincent Cheutet, Samir Lamouri. A PLCS framework for PDM / ERP interoperability. International Journal of Product Lifecycle Management, 2011, 5 (2/3/4), pp.295-313. 10.1504/IJPLM.2011.043182 . hal-00799829

HAL Id: hal-00799829

https://hal.science/hal-00799829v1

Submitted on 12 Mar 2013

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.

A PLCS framework for PDM/ERP interoperabilty

Thomas Paviot*

LCPI/Arts et Metiers ParisTech,
151, bd de l'Hôpital,
75013 Paris, France
E-mail: thomas.paviot@ensam.eu
*Corresponding author

Vincent Cheutet

Supméca/LISMMA,
3, rue Fernand Hainaut,
93407 Saint-Ouen, France
E-mail: vincent.cheutet@supmeca.fr

Samir Lamouri

LCPI/Arts et Metiers ParisTech,
151, bd de l'Hôpital,
75013 Paris, France
E-mail: samir.lamouri@ensam.eu

Abstract: Wide diffusion of methodologies and software relevant to Product Lifecycle Management (PLM) in industrial companies faces heterogeneity of information technology (IT) systems. Especially, the lack of interoperability between Product Data Management (PDM) systems, that drive virtual product development, and Enterprise Resource Planning (ERP), which manages real product, cannot lead to a global description of the product development process. We demonstrate that a mediator approach is pertinent for the coordination of these two systems. The use of open standards, and more precisely STEP Application Protocol 239, known as Product Lifecycle Support (PLCS), allows overcoming issues related to semantic part of this interoperability. The last part of the study focuses on implementation issues, in order to develop an interoperability framework based upon a consistent product data model, whether the viewpoint is from the design or logistics businesses.

Keywords: PLM;Product Lifecycle Management; Interoperability; PDM; Product Data Management; ERP; Enterprise Resource System; PLCS; Product Lifecycle Support; Standards; STEP; Standard for Product Data Exchange; SOA; Service Oriented Architecture.

Reference to this paper should be made as follows: Paviot T., Cheutet V. and Lamouri S. (2011) 'A PLCS framework for PDM/ERP

Copyright © 2011 Inderscience Enterprises Ltd.

interoperability', *Int. J. Product Lifecycle Management*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Dr. Thomas Paviot is graduated from the Arts et Metiers 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 focuses 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.

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.

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.

1 Introduction

The current trend of complexification in the development of new products in industrial companies leads to many issues. Complexification is understood here as the multiplication of partners (subcontractors, suppliers, customers located all over the world), the stronger integration between the actors of a project (design, production, marketing, maintenance etc.) as well as the huge amount of data produced, exchanged and stored during the development phase. Furthermore, although the economical environment is very constraining, the main goals remain to reduce the time-to-market, the costs, and improve the product quality. In this paper, we focus on companies in the field of aeronautics and defense industry, which can be characterised by the fact that (AIAA (2005)):

- the products (aircrafts, military vehicles, satellites) have long lifecycles (up to fifty years),
- they evolve in a context of strong regulation (security, traceability, long term data retention and preservation),
- these complex products require the participation of many specialized companies, some of them being considered as small and medium business enterprises,

- according to the nature of the project, many new partners may be included in the project team,
- some major companies can acquire other companies to get specific technologies,
- the IT system is highly heterogeneous due to economical or historical reasons.

The IT system that drives the product related data from design to retirement is then a strategic tool. The industrial deployment of this IT must satisfy the needs of three main type of actors:

- the design engineer, maintenance operator, manufacturer, salesman, must be insured that the coherency of the product data for its whole lifecycle,
- the IT manager must ensure the flexibility and the agility of the IT architecture, in order to allow a quick and efficient integration of the IT from a new partner,
- the head must reduce the total cost of ownership (TCO) of the IT, TCO including initial investment as well as maintenance.

In such industrial companies, the choice, deployment, use and maintenance of an efficient IT must then associate both of these actors. In a few words, the systems composing the IT have to interoperate at the lowest cost and with the best quality, i.e. ensure product data coherency. However the dispersion of information over heterogeneous models requires new solutions to preserve the overall coherence of the company information system (Roucoules et al. (2008)). The purpose of this paper is to demonstrate that the definition and the deployment of an industrial IT architecture as well as a suitable product model are deeply connected. The exclusive focus on one or the other point fails to achieve both goals previously mentioned. As a consequence, the paper is organized according to these two viewpoints: section 2 first introduces the PDM/ERP specific interoperability issue. Section 3 deals with interoperability general statements and then demonstrates the need of PLM standards, thus enabling to propose, in this section, a suitable service oriented functional architecture. Section 4 deals with the semantic issue of product representation and the way it can be overcame with the use of the STEP standard, especially the 239 Application Protocol known as Product Lifecycle Support. Section 5 proposes a framework, that instanciates the general architecture previously presented, and implements an AP239 based multiview data model validated with a simple scenario. The implementation is described in the section 6, as well as a study case. Section 7 finally concludes this work with the further work required to achieve the PDM/ERP interoperability.

2 PDM/ERP interoperability issue

2.1 Product Data Management tools and functional scope

The product design process is structured around two families of systems: authoring tools and Product Data Management (PDM) systems. Authoring tools are related

to creation/modification of design documents: 3D MCAD, ECAD, requirements etc. These authoring tools supply data to the Product Data Management (PDM) system. The PDM system is designed to manage product design data lifecycle. According to Audrain (2008), a PDM tool must have the following features:

- the exhaustive list of parts that composes the product,
- parts and documents maturity state management,
- integration with authoring tools,
- product breakdowns and multiview access (as-designed, as-built, as-planned, as-maintained)

However, PDM system is not intended to manage spare parts neither physical parts (nor parts as individuals), which is rather the role of the ERP system (see next subsection).

2.2 Enterprise Resource Planning tools and functional scope

According to Parr et al. (2000), ERP systems are comprehensive packaged software solutions which aim for total integration of all business processes and functions. Holland et al. (1999) report that ERP software automates core corporate activities, such as manufacturing, human resource, finance, supply chain management, sourcing, maintenance, orders/service forecasting, production planning by incorporating best practices to facilitate rapid decision-making, cost reductions, and greater managerial control. ERP is however not suitable for access design information, or to organize the technical data as well as engineering processes, althgough there is a narrow link between what has been developed during the design phase and how it is produced (Roucoules et al. (2008)).

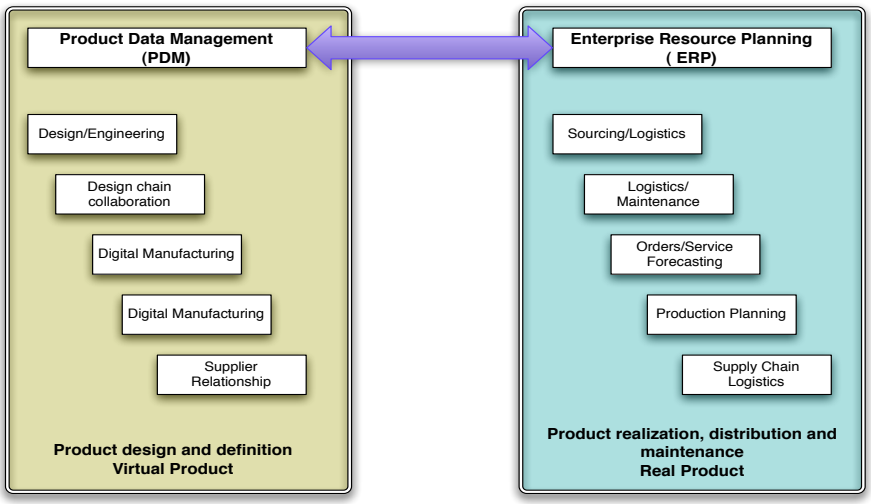
2.3 Need of an interoperability between PDM and ERP system

There is a deep relationship between the systems dedicated to design and production lifecycle stages: they must interoperate to enable information exchange and sharing between the design and production. Unfortunately there is no native integration or interoperability support between PDM or ERP commercial products, although the needs expressed by end users are well known: engineering bill-of-material (EBOM) to manufacturing bill-of-material (MBOM) bidirectional conversion, inventory level accessibility for designers, access to design information for maintenance operators and As-Maintained/As designed issues (Cimdata (2006)). One crucial step towards an ERP/PDM integration is then to build a semantic bridge between PDM parts/product breakdowns and production management concepts related to Material Requirement Planning (MRP), e.g. bill-of-materials, routings and work centers (cf. Figure 1).

2.4 Interoperability attempts in the PLM field

Although the PDM/ERP interoperability issue is depicted as opened, it is not wideley covered by the literature. Galeta et al. (2006) propose an *as-designed*

Figure 1 The need of a PDM/ERP interoperability



product model suited for the ERP, but this model deeply depends on a specific PDM software. They conclude this study by pointing out that the STEP PDM schema has to be considered rather than a specific data model. Xu et al. (2008) study the EBOM/MBOM issue, and define a set of transformation rules and 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 described. The main interoperability issue in the design phase seems to be the CAD/PDM integration and has been studied by Guyot et al. (2007), Oh et al. (2001) or Paviot et al. (2008). There are also a few works related to ERP/MES interoperability (see for instance Baïna et al. (2008)). However, according to the state of our work, it appears that the PDM/ERP interoperability field is still opened. These two families of systems involve people of different cultures that do not share the same semantic view of the product: while designers work to define a virtual product (i.e. a representation of the product using digital information) that fits the customer requirements, logistics focuses on the real product considered as a the incoming/outgoing piece of a set of time based activities (cf. Figure 1). This semantic issue is hereby considered as the major issue to be solved in order to achieve the PDM/ERP interoperability (as defined in the subsection 3.1).

3 Interoperability and standards

3.1 General statements about interoperability

According to the surveys of Kosanke (2005), Wegner (1996) and Panetto (2007), five different definitions of interoperability can be found. Among them, in

the following of this paper, interoperability will be defined as *the ability of two or more systems to exchange information and have the meaning of that information accurately and automatically interpreted by the receiving system.* This definition is particularly suited to the presented issue since it merges the technical and semantic concepts related to interoperability. When analyzed from the IT viewpoint, these close concurrent definitions can be merged in a multi-level interoperability approach (Euzenat (2001)). This author stands that five levels of interoperability must be completed in order to make an exchangeable representation understandable by two different systems: encoding (segment the representation in characters), lexical (segment the representation in words), syntactic (structure the representation in sentences), semantic (construct the propositional meaning of the representation), and semiotic (construct the pragmatic meaning of the representation, or its meaning in context). This typology can be compared to the study of EIF (2004), which defines three levels for interoperable enterprise systems: the technical level (data and message exchange), the semantic level (information and service sharing) and the organizational level (e.g. business unit, process and people interactions across organization borders). Considering products with long lifecycles, it's necessary to ensure the product data coherency over the time, up to decades. As a consequence, the product description must be robust whenever a change occurs in the enterprise organization and/or business processes. Due to the dynamic property of economical environment, these processes may be volatile whereas product information must remain stable over time. Baïna et al. (2008) introduce the concept of "product-oriented interoperability", i.e. the ability of different enterprise systems to manage, exchange and share product information in a complete transparency to the user. In that sense, the enterprise system interoperability issue must be driven by the semantic level, i.e. the product representation.

The use of standards is underlined in the literature as a way to help overcoming issues in the interoperability field. According to Chen et al. (2004) and Lee et al. (2005), enterprise interoperability requires a fair amount of open standards to be able to share and exchange information in a distributed organization and heterogeneous computer environment. Furthermore, there also are economical arguments in favour of standards. The American National Institute for Standards and Technology (NIST (2002)) identifies three costs of interoperability: avoidance costs, migration costs and delay costs. The NIST estimates that the use of the STEP standard (Pratt (2005)) has the potential to save up to one billion dollars per year by reducing interoperability costs in the automotive, aerospace, and shipbuilding industries.

The technical and economical considerations related to standards lead to investigate STEP as a possible way to stand as the basis of the *product driven interoperability* (cf. section 4) approach.

3.2 A mediator based interoperability architecture

When attempting to implement interoperability between two systems without changing them, two different approaches can be listed: a point-to-point integration or an Enterprise Application Interface (EAI) based interoperability. Assuming a set of n systems to connect, point-to-point integration requires a number of translators

equals to $n(n-1)$. This is thus unacceptable in terms of software development and maintenance costs (Guyot et al. (2007)).
The EAI solution, i.e. a central subsystem responsible for the communication between all other systems, is pertinent from an economical viewpoint since the number of required translators decreases to n. This approach has been recently extended with the "IT mediator" concept (Benaben et al. (2008)). Assuming the systems are Service Oriented Architecture (SOA) based, the mediator organizes data conversion, process orchestration and application management. This hypothesis is here validated since current trend is to make commercial products becoming SOA based (e.g. TeamCenter, Windchill, SAP, Adonix, MatrixOne etc.), and this approach is a possible and efficient way to consider PLM interoperability in general and design/logistics in particular. The 2.0 release of the OMG PLM Services (http://www.omg.org/spec/PLM/) provides a wide range of available services in the field of PLM: part identification, product structure, documentation management, shape/transformation, classification, properties, alias identification, authorization, configuration management, change and work management, process planning and a multi language support.
The product structure interchange and conversion part of this mediator is the core of this study. The next section of this paper explains how the use of standards can drive the work related to this topic.

4 Product model driven interoperability using Product Lifecycle Support standard

4.1 Standards in PLM

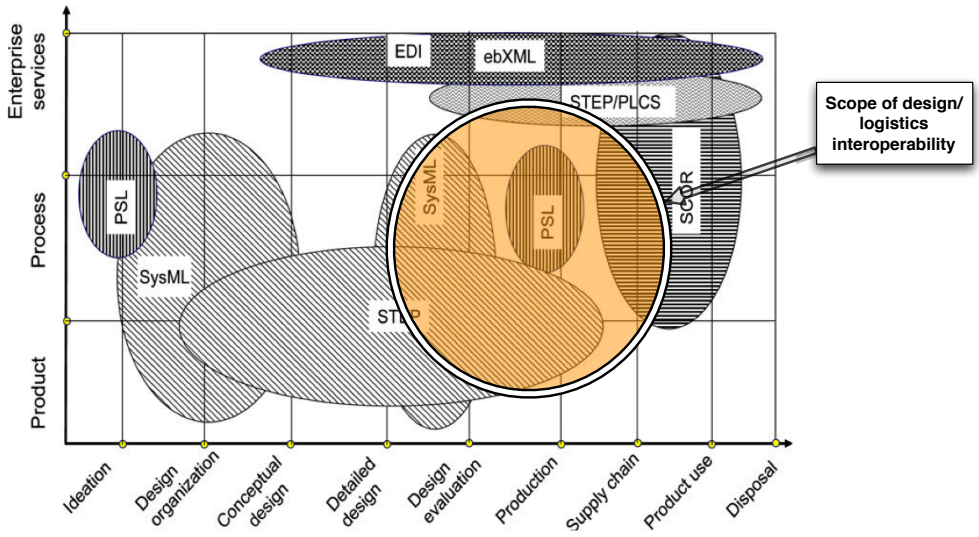
In a recent survey on standards in PLM fields, Rachuri et al. (2007) propose a four-level hierarchical typology of standards:

- Type 0: standards for implementation languages,
- Type 1: information modelling standards (like EXPRESS, UML, XML for instance),
- Type 2: content standards, domain of discourse: product information modelling and exchange standards (STEP), information exchange standards, product visualization standards, e-business and value chain support standards, security standards,
- Type 3: architectural frameworks standards.

The semantics part of interoperability is here adressed by the category of "Type Two" standards. According to this typology, the authors map the scope of the major current Type Two (content) standards of the PLM field along two dimensions (products life cycle major stages, complementary aspects of the information). They conclude that there is no standard that provides full coverage of the PLM support spectrum because it is fragmented and incomplete (cf. Figure 2). Considering the design/logistics interoperability context we focus in this paper, it is clear that there is no intersection in the Design/Production/Supply area that

could lead to a consistent product representation. In the next subsection, we focus on the way to overcome this issue.

Figure 2 Rachuri et al. PLM standards 2D map



4.2 The ISO 10303:239 or Product Lifecycle Support standard

The international standard ISO 10303 (Pratt (2005)) known as STandard for the Exchange of Product model data (STEP) covers computer-interpretable representation of product data, and its exchange. The objective of ISO 10303 is to provide a means of describing product data throughout the lifecycle of a product that is independent from any particular computer system. The really first STEP developments were lead by CAD data representation and exchange. STEP became after that a set of specific content standards known as Application Protocols (APs). At that time, STEP coverage scope did not permit to describe the whole lifecycle of a given product. It only allowed to structure specific industrial environment around one activity or product typology. Two major events then occurred in the early 2000s:

- In 2002, the STEP PDM Schema (Machner et al. (1998)) was released. It is a core set of entities in STEP that supports the mapping of concepts for Product Data Management (PDM), and built as a subset of APs 203, 214, 212 and 232. It is the first attempt of merging product semantics and data model from these differents APs,
- In 2005, the International Standard Organization released the AP239, known as Product LifeCycle Support (PLCS) (Rosen (2006)). PLCS is the first STEP Application Protocol addressing a wider scope than other APs and also proposing a generic and extensible product data model. This AP is thus discussed in detail because it is identified as a way to cover both design and logistics semantics.

AP239 offers a framework for the integration, exchange and management of technical data necessary for the support of a complex product and its evolution along its whole lifecycle (ISO (2005)). Its rich semantics offers information model required to define a complex product and its support solution, to maintain this product, and describe configuration change management of a product and its support solution.
PLCS is structured around the concepts of *product* and *activity*. In the PLCS semantics, a product can either be a product existing in the real world, a product that may come into existence as a consequence of some realization process, or a set of functions. As well, an activity is defined as the occurrence of an action that has taken place, is taking place, or is expected to take place in the future (change, distilling, design, a process to drill a hole are example of activities). PLCS data model then includes following descriptions:

- product modeling: representation of product assemblies, product through life and product history. This representation is based on the PDM Schema,
- process modeling: specification and planning of activities for a product, representation of the activity history of a product.

Aware that it would be impossible to fit all the specific needs of potential PLCS users, the steering comittee designed PLCS as en extensible standard (Rosen (2006)). PLCS thus offers a generic data model that can be specialized, under the condition of a methodology conformance, for specific business domains. This specialized data model is released in a Data Exchange Set (DEX) that defines how to use it. The OASIS consortium is responsible for the development and standardization of these DEXes. These statements lead to two observations for our purpose:

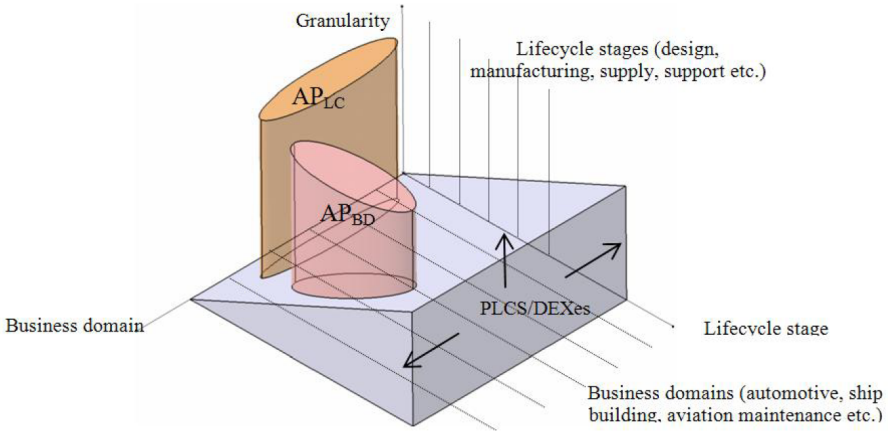
- on the Figure 2, the small axis of the ellipsis representing PLCS location in the (lifecycle, PPE) referential should be longer, since PLCS includes a rich and extensible data model for product representation,
- The scope of PLCS seems *a priori* suitable for PDM/ERP interoperability: the product representation, build upon the PDM Schema, match the PDM semantics whereas the activity model adds a temporal parameter that makes possible the description of design processes, manufacturing processes or routings. However, the Figure 2 shows figures out that PLCS is limited in scope to the enterprise services representation.

4.3 A graphical representation of PLM standards

The attempt of Rachuri et al. (2007) to map PLM standards to a two dimensional chart misses the fact that PLCS provides an extensible generic data model. This extensibility can be regarded in an other dimension related to business domain. A 3D map representation of STEP standards is proposed (cf. Figure 3), along the three following dimensions:

- lifecycle stage (LC) : this item defines the lifecycle stage of the product (e.g. design, manufacturing etc.),

Figure 3 3D Map representation of STEP Application Protocols



- granularity: some APs offer a more detailed product representation than others. Regarding all STEP APs, the wider is the scope, the less granular is the product model,
- business domain (BD): this item defines a specific business domains like automotive industry or ship building.

This representation shows that the PLCS data model, unlike other APs, can be extended to fit required granularity or business domain. The holes of Figure 2 can then be filled in with the appropriate data model specialization.

5 Proposed framework for PDM/ERP interoperability

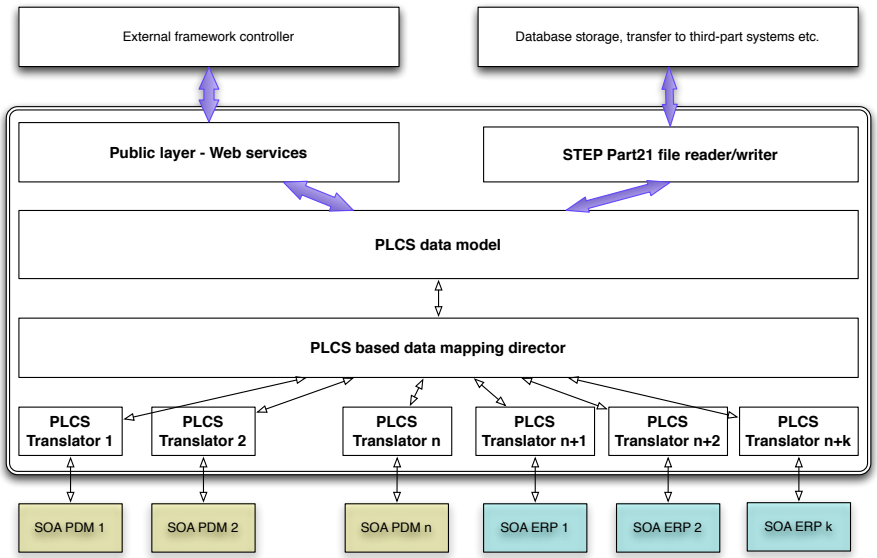
5.1 Functional architecture

This section presents the PDM/ERP interoperability framework (cf. Figure 4) based on a service oriented mediator (cf. subsection 3.2) and the PLCS product data model (cf. section 4). A first draft of this architecture was presented by Paviot et al. (2009). To avoid the redundancy of data, a volatile mapping between ERP and PDM objects is performed. Data flow and workflows orchestration are driven by business needs, i.e. fit the end-user of one system to grab from or populate data to the other system. This framework is composed of (from bottom to top on Figure 4):

- a set of PLCS translators, that convert the PDM or ERP requested object to or from a PLCS object,
- a data mapping director, which merges the information from the translators in a coherent PLCS description,
- a complete PDM and ERP semantic representation using the PLCS specialized data model. This data model feeds the data mapping whenever needed,

- a public layer intended to receive user queries and return appropriate information. This layer is the webservices part of the framework,
- a STEP file handler able to encode the product data into a STEP Part 21 file, as well as to retrieve information from such a file.

Figure 4 The PLCS/IT mediator based functional architecture



5.2 Technical architecture

The previous functional architecture is implemented according to the diagram presented in Figure 5. The PDM and ERP systems chosen are those available in the LISMMA laboratory: Windchill PDMLink/Project link as the PDM solution and OpenERP (http://www.openerp.com), a free and open source ERP system developped with the Python progamming language. The double lined box of the Figure 5 represents the boundary of the implementation part, which is composed of:

- an eXtended Markup Language/Remote Procedure Call (XML/RPC) based pipe for data transfer from/to OpenERP. OpenERP server/client built-in communication implements this protocol, and the server-side can easily be accessed with the Python xmlrpclib module to create parts, MBOMS, get information etc.,
- a SOAP based pipe from data transfer from/to the PDM enabling webservices access. Both PDMLink and ProjectLink offer a set of about 40 webservices to enable interoperability with third-part tools (workflow creation, parts and products creation, information extraction etc.). The

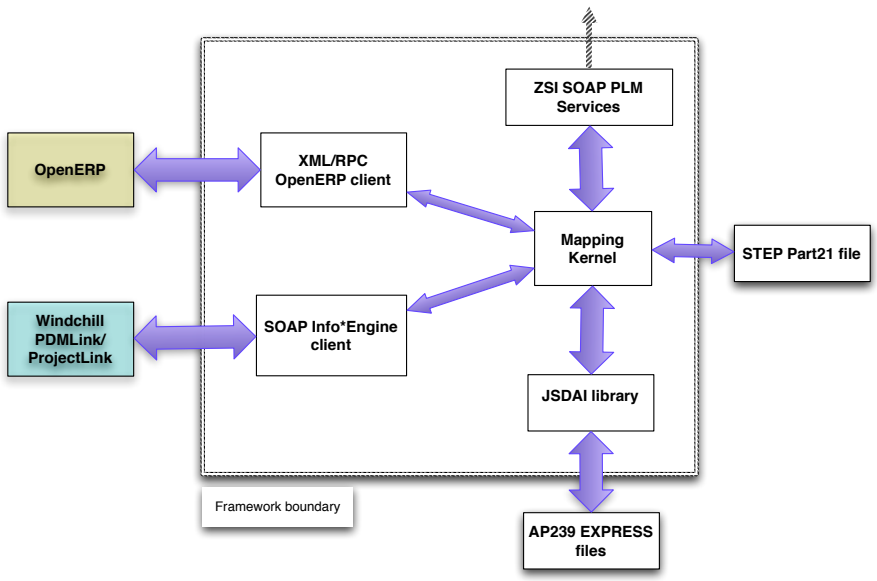
Zolera SOAP Infrastructure (http://pywebsvcs.sourceforge.net/), which is a Python library dedicated to webservices implementation, was used to parse the WSDL file and communicate with the system,

- the "Mapping Kernel" is intended to merge data coming from the systems and responsible for the consitency of design/production data. It is based upon an AP239 based data model whose implementation is detailed in the next section. The Python programming language is also used for the development of this part,

- handling of STEP AP239 objects (entities, types, functions, STEP file generation etc.) is performed with respect to the Standard Data Access Interface (SDAI) (ISO (1998)), that enables software applications to have dynamic access to a repository of data. The JSDAI library (http://www.jsdai.net) is a free and open source implementation of SDAI for the Java programming language, thus enabling reading, writing and runtime manipulation of object oriented data defined by an EXPRESS based data model. The EXPRESS file encoding the AP239 data model is parsed by this library which generates a set of Java classes. These objects are accessed through a java based Python implementation, known as Jython, to ensure consistency of the source code,

- at last, a set of webservices are implemented to give the user a total control over the behaviour of the solution.

Figure 5 Technical architecture



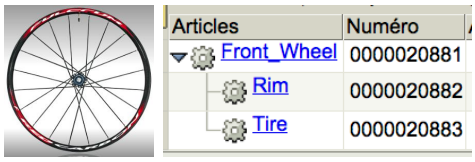
6 PLCS based implementation of the mapping component

In this section, the implementation of the "Mapping Kernel" is presented, the explanation being based on a simple test case presented in the following subsection.

6.1 Case study

In order to explain the implementation process, a very simple study case is defined hereafter: a bicycle front wheel, which is assumed to be composed of only two parts, one tire and one rim. Although this product could obviously be breakdown in more pieces, the purpose of this example is to be as simple as possible. The Engineering Bill of Material (EBOM) and Manufacturing Bill of Material (MBOM) are isomorphic: the EBOM defines a wheel as being composed of two parts in an organic view, where as the MBOM defines the assembly process of the rim and the tire to produce the wheel. As a consequence, although the EBOM and MBOM seem similar, they are quite different from a semantic viewpoint. Figure 6 shows the design view, as defined in Windchill Project Link, and the manufacturing view from OpenERP.

Figure 6 The bicycle front wheel simple study case - EBOM/MBOM



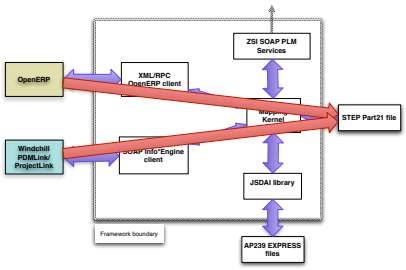
Two different scenarii are played upon this basic product: in the first one (identified as Scenario 1), the data coming from the PDM and the ERP are merged into a compliant PLCS based data model and a STEP Part21 file is generated. This file can then be used to populate empty or corrupted databases. In the second one (Scenario 2), EBOM is extracted from the PDM, mapped to a MBOM whithin the previous data model, and the ERP product structure is created from this MBOM. The Table 1 represents each scenario over the technical architecture (cf. Figure 5) and the associated algorithm (note that for the scenario 1, queries to PDM and ERP run in two separate parallel processes).

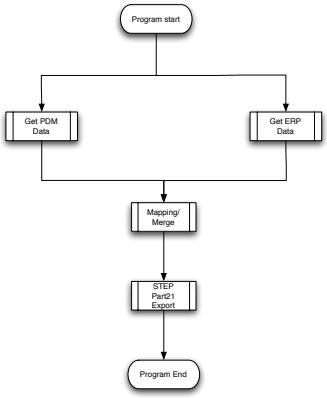
6.2 Implementation methodology

Although the generic STEP AP239 data model defined in the arm EXPRESS file can be used as-is, a set of rules or best practices are available to implementors. Conforming these prescriptions will ensure later that the implentation is compliant with the one from partners or suppliers. Indeed there are so many possible ways to implement PLCS that we must conform with those best practices. Those best practices are available from the PLCS dexlib project (http://www.plcs-resources.org) that defines a set of capabilities and templates. A *capability* provides usage guidance on how the PLCS information model should be used to represent a particular set of concepts, such as parts or properties. The *templates* have

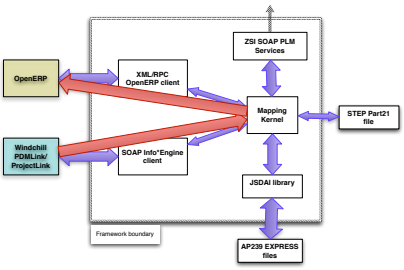
Table 1 Data flow/algorithm for each scenario (cf. Figure 5)

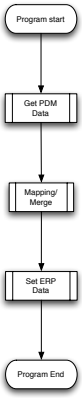
Scenario 1 :
PDM/ERP data merge and export





Scenario 2 :
transfer from PDM to ERP



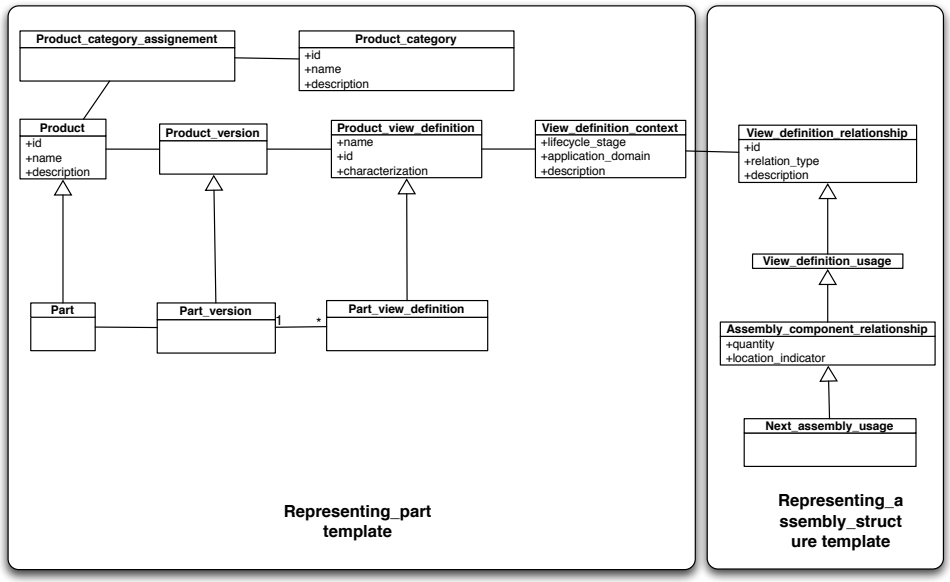


been defined to provide more precision to this usage guidance. They specify the exact instantiation of a collection of entities and attribute values. It is an unambiguous specification of how to use a collection of PLCS entities, attributes and reference data for a certain purpose within the context defined by a capability, thus ensuring consistent interpretation of use. To achieve the experiment, two specific templates are used and instanciated among the 45 available from the dexlib project: *representing_part* and *representing_assembly_structure*. The graphical representation of the related data model are given as an EXPRESS-G file for each template. The Figure 7 shows a UML diagram mapping this representation (Lubell et al. (2004)) (the reference data are omitted) in a consistent view (the entity **Product_view_definition** being the common component between those templates).

6.3 Multiview part representation

Windchill PDMLink/ProjectLink multiview features are quite simple: for a part, the view is defined as a metadata that can be set to “Design” or “Manufacturing”. According to the selected view, the corresponding product structure will be displayed. On the ERP side, OpenERP does not provide any multiview feature.

Figure 7 UML class diagram for the **representing_part** and **representing_assembly_structure** DEX templates



On the opposite, PLCS defines a rich semantic to describe the represented parts in a multiview context (the *representing_parts* capabilitiy/templatey was used). The **view_definition_context** EXPRESS entity, used to define views, define two attributes of the STRING type: **application_domain** and **lifecycle_stage**. That means a view, as defined in PLCS, has to be considered as a point (or a viewpoint) that can be represented on a 2D chart (cf. Figure 8), rather than a simple boolean state ("Design" or "Manufacturing") as defined in Windchill. For the bicycle front wheel scenario presented above, two view points are defined:

PDM_VIEW_CONTEXT = ("design", "mechanical_engineering")
ERP_VIEW_CONTEXT = ("manufacturing","production")

The second step is then to instanciate two **Product_view_definition**

6.4 Product structure representation

The product representation data model of PLCS is much more complete that the PDMSchema, used for instance by Tursi et al. (2009) to structure the product. The *representing_assembly_structure* capabilitiy/templatey was used.
The STEP Part21 outcoming file generated by the framework, for the related scenario, is shown on the Figure 9 .

Figure 8 AP239 multiview representation

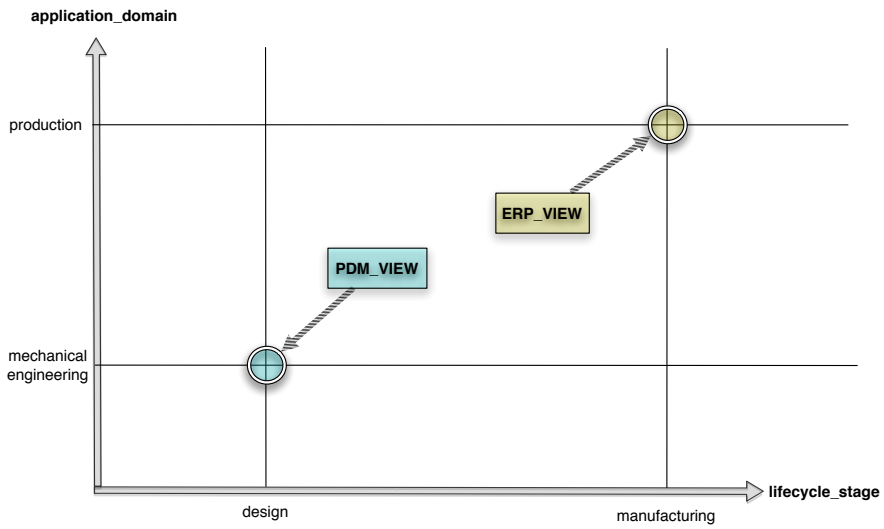


Figure 9 STEP Part21 file for the bicycle wheel sample

```
ISO-10303-21;
HEADER;
/* Generated by software containing
 * JSDAI (TM) from LKSoft (www.lksoft.com, www.jsdai.net)
 * JSDAI Runtime Version 4.0.0 (Build 270, 2009-07-22T20:05:26)
 */
FILE_DESCRIPTION(
/* description */ ('Bicyle font wheel sample AP239 p21 file'),
/* implementation_level */ '2;1');
FILE_NAME(
/* name */ ' ',
/* time_stamp */ '2009-12-13T22:45:52',
/* author */ ('Thomas Paviot','Vincent Cheutet','Samir Lamouri'),
/* organization */ ('Supmeca/LISMMA'),
/* preprocessor_version */ ' ',
/* originating_system */ 'JSDAI MULTIPLE Version 4.0.0 (Build 270, 2009-07-22T20:05:26)',
/* authorization */ 'Thomas Paviot');
FILE_SCHEMA(('AP239_PRODUCT_LIFE_CYCLE_SUPPORT_ARM_LF'));
ENDSEC;
DATA;
#1=PART('id_tire','tire',$);
#2=PART('id_rim','rim',$);
#3=PART_VERSION('the_id','optional description',#1);
#4=VIEW_DEFINITION_CONTEXT('mechanical_engineering','design',$);
#5=VIEW_DEFINITION_CONTEXT('production department','production',$);
#6=PART_VIEW_DEFINITION('pdm_view_id',$,$,#4,$,$);
#7=PART_VIEW_DEFINITION('erp_view_id',$,$,#5,$,$);
ENDSEC;
END-ISO-10303-21;
```

7 Conclusion and further works

The PDM/ERP interoperability is still an issue for most industrial companies, especially those dealing with long lifecycle products. However, an approach based on open standards can overcome these economical as well as technical issues. The use of the STEP PLCS standard can reconcile the design and logistics viewpoints. The framework proposed in this study transfers product data from PDM to ERP (or the opposite), and can also merge data coming from these systems and export them to a compliant STEP Part 21 file. However this basic implementation still misses essential features related to other issues: the management of product diversity, changes and propagations mangement (the EBOM and MBOM being dynamic objects), as well as a more advanced mapping technique/algorithm able to handle a wide variety of product structures.

References

Aerospace Industries Association of America, Inc (2005) 'The future of aerospace standardization', http://www.aia-aerospace.org/assets/aerospace_standardization0105.pdf.

Audrain, P. (2008) 'Management of industrial assets: integration of a PDM software with an EAM tool', *9èmes Etats Géneraux de Micado*, Cachan, France.

Baïna S., Panetto H. and Morel G. (2008) 'New paradigms for a product oriented modeling: Case study for traceability', *Computers In Industry*, doi:10.1016/j.compind.2008.12.004.

Bénaben F., Touzi J., Rajsiri V., Truptil S., Lorré J.P. and Pingaud H. (2008) 'Mediation Information System Design in a Collaborative SOA Context through a MDD Approach', *Proceedings of MDISIS 2008*, pp.89–103.

Chen, D. and Vernadat, F. (2004) 'Standards on enterprise integration and engineering - A state of the art', *Int. J. Computer Integrated Manufacturing*, Vol. 17, No. 3, pp.235–253.

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

European interoperability framework (2004), white paper, Brussels, Belgium.

Euzenat, J. (2001) 'Towards a principled approach to semantic interoperability', *Workshop on Ontologies and Information Sharing*, Seattle, Washington, USA.

Galeta T., Kljajin M. and Karakasic M. (2006) 'Product model suited for the erp system', *International Design Conference - DESIGN 2006*, Dubrovnik - Croatia.

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

Holland C.P. and Light B. (1999) 'A critical success factors model for ERP implementation', *IEEE software*.

ISO 10303-22 (1998) 'ndustrial automation systems and integration – Product data representation and exchange – Part 22: Implementation methods: Standard data access interface'.

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

Kosanke, K. (2005) 'ISO Standards for Interoperability: a comparison', *1st International Conference on Interoperability of Enterprise Software and Applications*, Geneva, Switzerland.

Lee S. and Jeong Y. (2005) 'A system integration framework through development of ISO 10303-based product model for steel bridges', *Automation in Construction*, Vol. 15, pp.212–228.

Lubell J., Peak R.J., Srinivasan V. and Waterbury S.C. (2004) 'STEP, XML and UML: complementary technologies', *Proceedings of ASME 2004 Design Engineering Technical Conference*, September 28-October 2, Salt Lake City, Utah USA.

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

National Institue for Standards and Technology (2002) 'Economic Impact Assessment of the International STEP Standard in Transportation Equipment Industries', http://www.uspro.org/documents/STEP_study.pdf.

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

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, doi 10.1007/s00170-002-1454-4.

Panetto, H. (2007) 'Towards a classification framework for interoperability of enterprise applications', *Int. J. Computer Integrated Manufacturing*, Vol. 20, No. 8, pp.727–740.

Parr A.N. and Shanks G. (2000) 'A Taxonomy of ERP Implementation Approaches', *33rd Hawaii International Conference on System Sciences*, Hawaii..

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.

Rachuri S., Subrahmanian S., Bouras A., Fenves S., Foufou S. and Sriram R. (2007) 'Information sharing and exchange in the context of product lifecycle management: Role of Standards', *Computer-Aided Design*, doi:10.1016/j.cad.2007.06.012.

Rosen J. (2006) 'Federated through life-cycle support', *1st Nordic conference on PLM, NordPLM 06*, Göteborg, Sweden.

Roucoules, L. and Noel, F. (2008) 'The PPO Design Model With Respect to Digital Enterprise Technologies Among Product Life Cycle', *Int. J. Computer Integrated Manufacturing*, Vol. 21, No. 2, pp.139–145.

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, pp.238–245.

Wegner, P. (1996) 'Interoperability', *ACM Computing Survey*, Vol. 28, No. 1, pp.285–287.

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.