An MDA Approach for PLM System Design

Onur Yildiz, Lilia Gzara, Philippe Pernelle, Michel Tollenaere

An MDA approach for PLM system design

Onur Yildiz[1,2], Lilia Gzara[2], Philippe Pernelle[3], and Michel Tollenaere[2]

[1] Audros technology
F-69003 Lyon, France, 41 rue de la cité
oyildiz@audros.fr
[2] INP Grenoble, Laboratory G-SCOP
F-38000 Grenoble, France
lilia.gzara@grenoble-inp.fr
Michel.Tollenaere@inpg.fr
[3] University of Lyon 1, Laboratory DISP
F-69621 Villeurbanne Cedex, France
philippe.pernelle@univ-lyon1.fr

Abstract. Design and reconfiguration of industrial information systems is an important issue for SME/SMI. In this context, these small enterprises need tools and methods that allow them to adapt their system to their business while ensuring its consistency. The aim of this paper is to provide an approach for both the design method and the control method. This approach for PLM (Product lifecycle management) system is based on MDE (Model Driving Engineering). First, we will present a metamodel for PLM and the model transformation concepts. Then, we will present an application with the Audros PLM system.

Keywords: PLM system, MDE, MDA, metamodel

1 Introduction

In recent years, many SME/SMI have structured their information system on PLM systems. Their initial goal was to optimize the development process. Thanks to customization capabilities of PLM systems, these SME/SMI have configured their own product models based on their business activity. In addition to these changes, they have mostly added automation tasks, in order to optimize their business processes. All these configurations should not be an obstacle to the flexibility which is nevertheless one of their major assets.

In this context the main problem for its businesses, is to have a simple means to manage adaptations. In this way, the necessary system reconfiguration is constrained by a double consistency. The first is structural consistency. This is to keep consistency between the rules for structuring models (Product and Organization). For example, the business views trades use attributes of the data model, users rights depend on the attributes of the classess trades. This structural correlation implies dependence in case of modification of the company's customer model. Thus, the suppression of a class object's attribute should generate its own suppression in all the business views.

The second relates to the process consistency. To this end, it is necessary to verify impacts of process changes on structural models (and vice versa). In some cases, the management of trades processes is distributed between the Audros PLM and other tools. In other cases, a process managed in the PLM can call upon points in external processes managed in tools. For example, business management in the PLM may require a call to points in external processes managed in some other tools like ERP (Enterprise Resource Planning), to start the order process of some components identified during the study of the business or to start the invoicing process of the business. These interactions imply the transmission between the PLM and ERP of certain data (business data, components data, customers data).

In this paper, we present a design or reconfiguration approach of PLM systems around MDE (Model Driving Engineering). Indeed, the main advantage of PLM systems is to adjust to business needs. These needs (data customization) exist in all cases and whatever the company size, but it increases the implementation time. In fact, many objects are handled from the same concepts. This has motivated our approach. In section 2, we reiterate the fundamental principles of the MDE approach (more particularly MDA) by proposing a metamodel. Then in the next section we illustrate model design from metamodel and instantiated in the PLM system Audros

2 Design his PLM by MDE/MDA approch

2.1 Model Driving Engineering

Model Driving Engineering (MDE) [1] was at the origin of improvements in complex system development. It consists in a form of generative engineering or all of a part of an application is generated starting from models. There are two key stages in the engineering directed by models: the definition of the DSM (Domain Specific Modelling) and the transformation of these models. Contrary to the object approach, which is based on instantiation and inheritance relations, engineering directed by models uses different relations concept : "RepresentedBy" and "ConformAt".

The second key phase of the IDM is the transformation of model. It allows to make models operational, to transform a source model into another target model of the same system but on a different level [2]. We can identify two types of transformations:

- Endogenous transformation from the same meta-model, the source model and the target are in conformity with the same meta-model
- Exogenic transformation starting from different meta-models, the source model and the target are not in conformity with the same meta-model.

It is on these general principles that the OMG (The Object Management Group) is based to define its standards. With OMG's vision, the abstraction level to a meta level is limited. OMG set up a language of definition of metamodels which has the form of a model, the MOF (Meta-Object-Facility) [3]. So to avoid a too great number of abstraction levels, the MOF has the capacity to describe itself. The MOF is at the top of architecture on four abstraction levels for modeling systems presented in the following diagram :

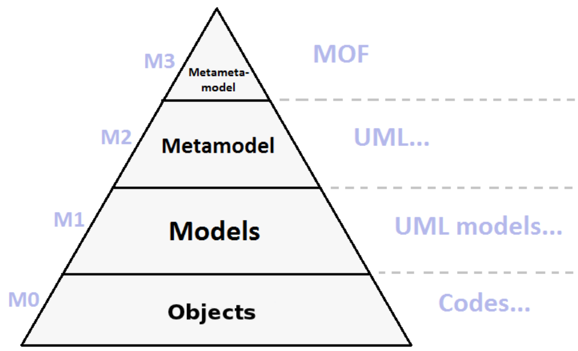

Fig. 1. modeling pyramid by OMG

MDA (Model Driven Architecture) [4] [5] is an approach to system development provided by the OMG and using models, metamodels, and meta-metamodels. MDA proposes three perspectives (CIM : Computation Independent Model, PIM : Platform-independent Models, PSM : Platform-Specific Models) to design models.

The CIM is independent of any information system. It allows the system vision and helps to represent what the system will do exactly. It is useful to understand a problem as a source of vocabulary shared with other models. The technical independence of this model allows it to keep all its interest over time and is changed only if the knowledge or business model needs change.

The PIM is independent of any technical platform (EJB,. NET ...) and does not contain information on technologies that will be used to deploy the application. This is a model that represents a partial view of an CIM. It shows the operation of entities and services. It describes the system, but does not show the details of its use on the platform.

The PSM is dependent of the platform used. It is used for code generation. The PSM describes how the system will use these platforms. There are several levels of PSM. The first, derived from the transformation of a PIM. The others are obtained by successive transformations. They allow to obtain the code in a specific language. In our context, the PSM level stops at an execution level within a PLM system.

The idea is to be able to manage heterogeneous models independently from any platform (PIM) and to pass by transformations to specific models depending on selected platform (PSM) and to finish with specific elements (table structure, scripts,...) executed in a PLM system.

Fig. 2. OMG's workflow

Our work is on levels M2 (metamodel), M1 (business model), M0 (PLM implementation model)

2.2 A meta-model for PLM

There is no one standard metamodel for PLM. There are several proposals which are related to the structural dimension of the product [6] [7]. In this paper, the first approach is an approach to design and reconfiguration. The metamodel is one element that allows to develop consistent models.

Our proposal is leaning on an MDA approach. In this approach, we propose a generic meta-model (M2) that describes the CIM viewpoint (Computation Independent Model). The meta-model is the result of a abstraction process of different models/frameworks proposed [8] [9] [10]

These are the concepts for industrial products development (application domain).

- the actor is a person or an organizational element who has expertise in business or in information system.
- the business object is the main type of concepts that is readily understandable by all business stakeholders. A business object is a bearer of product information and must be unambiguous.
- the IS resource is a concept that is used by the information system designer
- the business properties characterize a business object. These properties are defined on quantifiable domains.
- the IS property is a design element that is necessary to any PLM (eg in a PLM system any object has an unique ID)
- the functional link is a dependency between business objects

These concepts are represented in the following class diagram 3

2.3 PIM and PSM viewpoint in PLM systems

A PIM must have a sufficient degree of independence in order to enable its mapping to more platforms. In our context, it relates to be independent of the

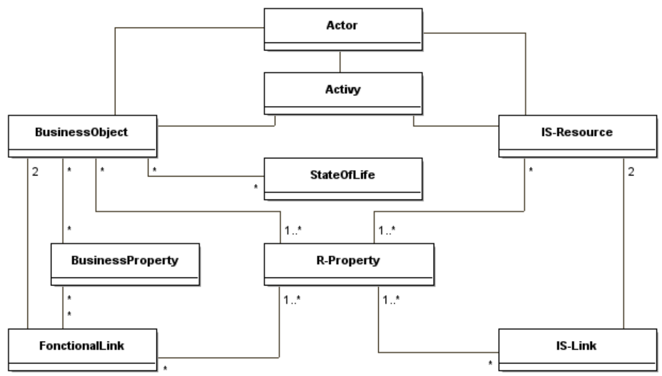

Fig. 3. metamodel from CIM viewpoint

various PLM systems. The business model is based on stereotypes characterization from the CIM metamodel. Thus, the resulting model will be in accordance with previous metamodel. It represents the business concept that will be handled in the enterprise. The compliance control is ensured by a set of constraints (eg all business object should possess at least a system property and a business property).

For example, enterprises working with CAD documents, a PIM model is given by the following diagram (Fig. 4)

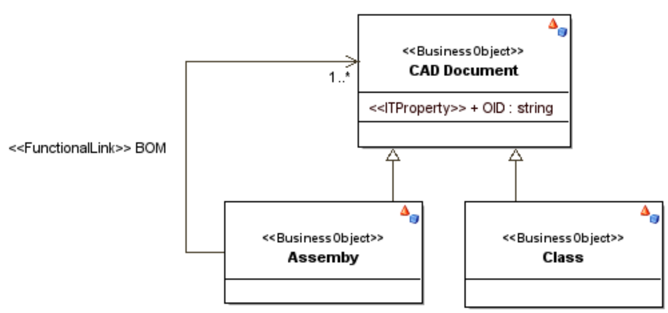

Fig. 4. Example 1 of PIM model

Another example is the viewpoint notion that is present in all PLM systems and that allows to characterize the access rights on the business objects (Fig. 5)

PSM level is the level of implementation models. In our case, the PIM concepts transformation does not produce executable code. Indeed, our approach

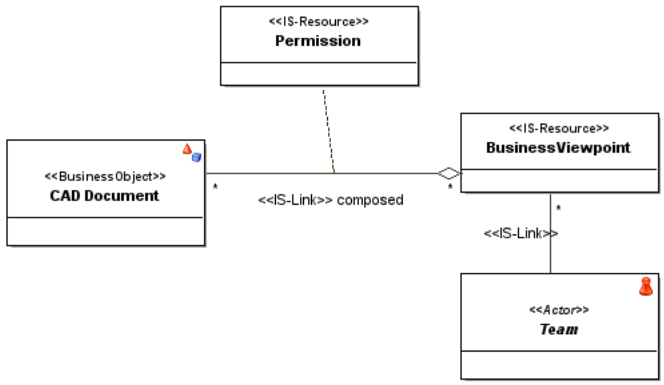

Fig. 5. Example 2 of PIM model

is intended to be used in a PLM context (or reconfiguration definition). The execution platform is implicitly a PLM system and generated elements are in fact elements that will be instantiated in the PLM by the application server.

The purpose of PDM (platform definition model) is precisely to describe the templates for the models transformation (from PIM to PSM).

In the next section, we illustrate a business model (PIM) within a PLM platform (Audros) which is very used in SME/SMI.

3 Application with an industrial PLM context

3.1 Use case with Audros PLM system

Audros is a PLM system which is deployed mostly in SME/SMI. His architecture model breaks up into various elements allowing data management, business views and users rights management. These elements are in strong correlation.

From these constraints and the metamodel, we built the Audros model Audros (the 'Standard Edition'). This model is designed for small business who want to quickly deploy a PLM system

3.2 Modeling example : Audros's Standard Edition (SE)

The SE model is proposed by Audros for its PLM system. The objective of SE is to propose a minimalist model that works for an SME . it must contain :

– useful objects (Part, Product, Document, CAD doc, BOM, ...)
– standard viewpoints (design, engineering , manufacturing, ...)
– standard change management (ECR, ECO, ...)

The following diagram (fig. 6) shows an extract from the SE's PIM model. In this model, the metamodel concepts are represented by stereotypes (UML2.0)

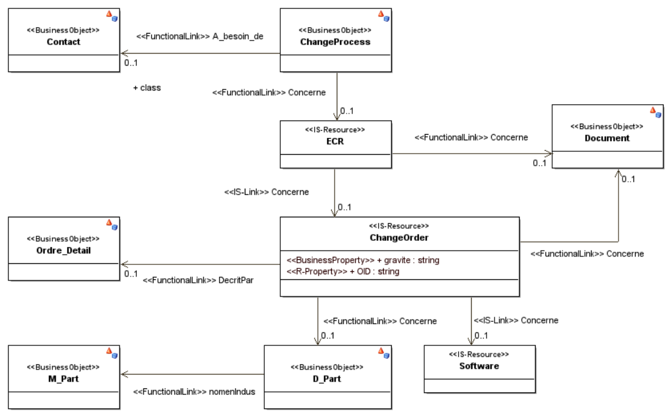

Fig. 6. Extract from PIM model

At this level the model is not dependent on the Audros platform, compliance is checked by the metamodel and constraints.

In the next step, the creation of the elements in PSM viewpoint is using templates and transformation rules. The PSM elements have to be executed in Audros either under structural shape (Database) or under dynamic shape (ScriptAUPL). About structural aspects, the transformations of a *BusinessObject* or an *ISRessource* is realized from transformation relations (BusinessOjbectToAudrosCTable, BusinessPropertyToAudrosCColumn, ISPropertyToAudrosCColumn, ComplexAttributeToColumn, ...)

For example, in Audros's SE, the transformation of part and BOM is given by

$$M\_PART, D\_PART, BOM \rightarrow \begin{pmatrix} 2\ tuples\ in & ClassAudrosTable \\ 1\ tuple\ in & RelationAudrosTable \\ 3\ tables & M\_PART\_Table \\ & D\_PART\_Table \\ & BOM\_Table \\ 1\ tuple\ in & BOM\_Table \end{pmatrix}$$

Changing needs is carried on the PIM level. Such changes, modifications, additions or deletions must meet the metamodel.

4 Conclusion

The goal of this paper is to verify the pertinence of an MDE approach in design and reconfiguration of a PLM system. With the metamodel proposed, we define a set of minimalist concepts for the control of conformity. With templates and transformation rules, we validated the feasability of a generation in an industrial system for SME/SMI. Currently constraints and transformation rules are not exhaustive especially for dynamic aspects of the model.

Moreover, in the case of Audros, automated jobs can modify data depending on users events (for example: BOM creation following an event Project creation). These tasks are carried out by AUPL scripts (AUdros Programming Language).These can modify the data model or interact with it. To manage these various dependences and to maintain coherence a total of the system, Audros would need mechanisms to control the dependences (for creation and modification) between the various elements of its structure, as well as mechanisms of synchronization of scripts respecting the modifications of the model. The ideal would be to lay out a tool which detects modifications and propagates their impacts within the model and scripts.

References

[1] Favre, J.: Concepts fondamentaux de l IDM. (2005)

[2] Combemale, B.: Approche de meta-modelisation pour la simulation et la verification de modele. Application a l ingenierie des procedes. PhD thesis, Institut Polytechnique de Toulouse (2008)

[3] OMG: Meta object facility (mof)[online]. http://www.omg.org/spec/MOF/2.4.1/ (2011)

[4] OMG: Omg model driven architecture [online]. http://www.omg.org/mda/ (2001)

[5] Bezivin, J., Gerbe, O.: Towards a precise definition of the omg/mda framework. In: 16th IEEE International Conference on Automated Software Engineering, San Diego, USA (November 2001)

[6] Srinivasan, V.: STEP in the context of Product Data Management. In: Advanced Design and Manufacturing Based on STEP. Springer (2009)

[7] Abid, H., Pernelle, P., Noterman, D., Guillmot, M., Ben Ama, C.: Integration of mechatronic products within a plm system. In: INCOM. (2012)

[8] Sudarsan, R., Fenves, S., Sriram, R.: A product information modeling framework for product lifecycle managemen. Computer Aided Design **37**(13) (2005)

[9] Chettaoui, H.: Interoperabilite entre modeles heterogenes en conception cooperative par des approches d'Ingenierie dirige par les modeles. PhD thesis, Institut polytechnique de Grenoble (2008)

[10] Eynard, B., Gallet, T., Roucoules, L., Ducellier, G.: Pdm system implementation based on uml. Mathematics and Computers in Simulation **70**(5-6) (2006) 330–342