Managing the evolution of product configuration within PLM systems

Seyed-Hamedreza Izadpanah, Lilia Gzara, Michel Tollenaere

Managing the evolution of product configuration within PLM systems

Seyed-Hamedreza IZADPANAH, Lilia GZARA, Michel TOLLENAERE.

University of Grenoble
G-SCOP Laboratory, 46, Ave. Felix Viallet, 38031 Grenoble Cedex 1, France
Email: **Seyed-Hamedreza.Izadpanah@g-scop.inpg.fr**, **Lilia.Gzara@g-scop.inpg.fr**, **Michel.Tollenaere@inpg.fr**

Abstract: A critical aspect of PLM systems is their product information modeling architecture. Therefore, dealing with the product structure model and its evolutions or transformations plays a significant role in the development, adaptation and performance of PLM tools. This paper deals with the product structure model changes. It supposes to formulate a transformation function to allowing the evolution of a model such as the design model of a product line. In order to realize this objective, the joint evolution of generic product models and specific product models is studied, based on the principles of the MDA (Model Driven Architecture) approach, the transformation function is defined.

Keyword: Product Lifecycle Management, Product Structure Model, Model-Driven Architecture, Evolution of Information System.

1 Introduction:

Product structure model is used to explain the decomposition or breakdown of a product in a specific phase of its lifecycle, according to the requirements of that phase. Therefore, there exists a variety of product models which describe a single product with different approaches. This set of models necessitates setting up a framework in order to validate any change, modification, transformation or evolution of the models. This framework should make product models compatible and interpretable [Eynard 06].

1.1 Evolution of PLM applications

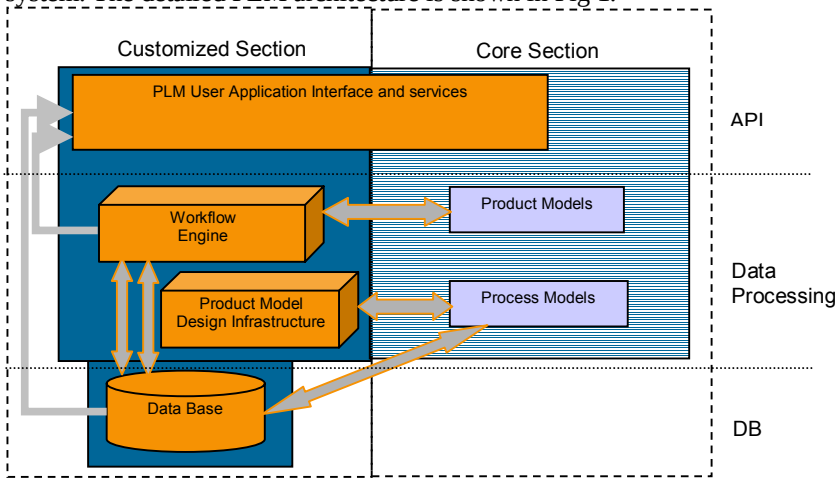
There are many reasons that motivate an enterprise to evolve or modify its information system such as PLM application. Before going to detail, it should be noticed that the origin of this evolution determines the parts of PLM which should be changed. Furthermore, the procedure of evolution is chosen, based on this origin. The evolution of the enterprise offers, the change in its organization or modification of its operations, are the examples of evolution origins. In all cases, the existing PLM application may be

evolved and consequently, the data already stored through the ancient application must be reformed upon the new models.

The effort to make the PLM more compatible with other management or planning tools inside or outside the enterprise is another drive. The improvement of compatibility and interoperability between the different information systems inside the enterprise, such as ERP or other applications used for design, manufacturing, etc. is considered as the objectives of IS services. Moreover, communicating with the IS of other related enterprises such as suppliers or clients is remarkable. To achieve it, the evolution of some parts of the existing IS may be inevitable.

Migration between various PLM applications is possible. A company that is joined with other companies is obliged in the most cases to changing or revising its IS, especially its PLM.

PLM applications, like other information systems, consist of different parts or layers which constitute an architecture. With the broadest view point, this architecture can be divided into two general sections: 1-the part that is provided by the PLM vendors (the toolkit before the customization), and 2-the other part that contains all the customization and adjustment which has been done on the system to integrate it into the enterprise information system. The detailed PLM architecture is shown in Fig 1.



Error! Bookmark not defined.

Figure 2, PLM Application Architecture

As mentioned before, the evolution strategy varies, depending on the part of PLM system that should be evolved. This part depends on the evolution cause. Then if a new product is going to be launched and it is obligatory to modify the model or meta-model of the product, therefore the evolution return to a product model transformation problem. However, if the organization or the process is under modification, the problem is more a changing of the workflow part. Moreover, the modification of the interface or the links between the different parts of the PLM architecture can be followed by different motives of evolution. Last but not the least, the core sections of PLM application, (that is PLM system) may be under modification if the software of PLM is changed or some modules are added or modified by its editor. Hence the product model should be revised in order

to perform the possible modification. In this paper, we only deal with the evolution of the product model.

1.2 The importance of the product configuration in the evolution of PLM applications

Product configuration, i.e. the set of functional and physical features of a product, is the principal concept in PLM systems. PLM systems offer a set of mechanisms in order to support product configuration management. These mechanisms include the documents and BoMs (Bill of Materials). BoMs are instantiations of one or more product models in a database. Therefore the role of product model within the PLM is to structure all product-related information and their connections [Schuh 06].

In this context, any modification in the product model plays a fundamental role in the evolution of the whole PLM application. Product model evolution means modification of the model architecture, its elements and their relations. In the object-oriented modeling approach, it means the modification of classes, attributes, associations, and methods.

It should be noted that here, the term of "product model evolution" may be used for explain two concepts. The one that is about the evolution of product model during its life-cycle, which means the modification of model caused by the different process of design, manufacturing, etc. It can be cited that this modification is done on the instantiation of product model. But the second one that is under the study in this paper is regarded to modification of product model in one stage of its life-cycle. In this case, the product model is changed, not because of the design process requirements, but in the reason of evolution in the information system or the enterprise organization.

These items show the importance of the product model evolution analysis in relation to the PLM evolution. In fact, during the procedure of PLM evolution, the most delicate subject that can affect the whole procedure is how product model should be modified. Moreover, as mentioned before, some of PLM evolution causes are related to product model modification itself.

As mentioned before, product model evolution returns to the problem of model transformation. In this context, one of the major questions is to determine if the implementation of model transformation should be started in the meta-model level or if the transformation is not related to the upper (meta-model). Each PLM system uses its own structure to design the product model, i.e. the meta-model allowing to build the various product models are different in the PLM system. In the case where the PLM system itself is changed, it should be firstly ensured that the meta-model transformation is necessary or not. In the other hand, in the case of model transformation with the same meta-model, the procedure of model transformation probably conforms to the meta-model.

1.3 The paper outline

The rest of this paper is divided into three sections. First, section 2 deals with product models evolution. Then, section 3 tackles the problematic of model transformation, from a theoretical and an applied point of view. Finally, section 4 specifies the first elements of a function allowing transformation of product models.

2 Product Models evolution

Evolution of product models within an information system is caused by two principal reasons:

- The first reason is due to the differences existing among models generated by actors with different business viewpoints such as manufacturing, design (with different fields such as electric, mechanic, and thermal) and assembly. In this case, different models co-exist to represent different aspects of identical objects. Therefore the mechanism of transformation among the existing models used in the enterprise during the product lifecycle may be interesting.
- The second reason belongs to the evolution of the information system used within these business viewpoints. As mentioned before, there are plenty of reasons that push to evolve or modify existing product models. The modification of the enterprise organization, the change of the products offered by it, or the evolution of its information system can be the sources of product model evolution. Moreover the advancement or replacement of the PLM system used by the enterprise can be another reason for the evolution of product model.

Product models evolution was the subject of many researches. They can be categorized into two types:

- Works related to the evolution of product model during the design process and to the synchronization of product model between different business viewpoints.
- Works related to the evolution of technical data, among different product modeling languages. In the other word, these researches are interested to study the process of migration of instances of models, and standardize the rules of data exchange between different languages. These works are not completely related to product modeling, but they are interesting because of two reasons. First, after the model transformation, migration of data must be done. This migration requires these methods. Second, these methods are about the evolutions in modeling framework.

It should be noted that some efforts has been done to do mapping between the models in the different systems, such as PDM, CAD and CAM. [Oh 01], or mapping among the diverse representation language or schema of models and meta-models. [Krause 07] In these cases, the transformation between models is not always considered, even though the identification of mapping specification between models is a question of model transformation.

Based on these perspectives, the studies already have been done will be analyzed. Sudarsan et al. proposed a generic framework for the design of product information modeling architecture. This research was done within a NIST project in order to make into coherence the different product configuration during the lifecycle. [Sudarsan 05] Eynard et al. have paid attention to the different breakdown business-viewpoints such as assembly or manufacturing and tried to avoid duplicating of the data stored in the PLM by recommending the mixed modeling structure. [Eynard 06] and [Eynard 04] Sevensson discussed the basic technical requirements for the Product Structure Management strategy. He highlighted the capabilities to create and manage the product models, keep the history of all of the activities around the product and ensuring the traceability and consistency of the information. Finally he proposed a master structure from which all of the views can be derived. [Svensson 02] Gzara et al. proposed a method for building a

product model by reuse of patterns. Based on a set of rules and criteria, the product model is built progressively by assembling different model fragments obtained by reuse of patterns [Gzara 03] Eastman et al. analyzed the evolutionary product model development with the viewpoint of database languages. They choose a suitable type of model mapping which enables following and handling the evolution of product model during its lifecycle. Then they proposed a language capable to facilitate the mapping, derivation of views, etc. [Eastman 99].

Zina et al. also worked on the generic modeling of the product. Regarding to product modeling, they divided the approaches which were used to take into account the actors' viewpoints to two types, multi-view and multi-model. The multi view approach is based on the development of a single model starting from different views. Therefore there is only one unique generic model, and the derived models are the writable (it is possible to change or modify the views, and this modification will be made also in model generic) view of it. In the other hand, in the multi-model approach, there are many models; each one is conformed to one business viewpoint. In this optics, the coherence rule is unavoidable. [Zina 06]

Regarding to second aspect, which deals with the evolution of existing product model, no research has been observed in the papers. The problematic is important as it will be studied in the industrial case.

3 Model transformation

3.1 The different types of product model transformation

The product models stored in PLM tools can be categorized in three levels: the generic product, the specific product and the exemplary product [Gzara 03]. Generic product is a virtual model that describes a line of products. For instance, a model which describes the line of cars "Peugeot 206" is a generic model. Each generic model can contain several sub-categories which describe the different types within each line (for instance: Peugeot 206 HDI 1.6, Peugeot 206 SW 1.2 …). These sub-categories are specific product models or product-type models. The exemplary product is the description of a physical product which is fabricated in the company (for instance: my Peugeot 206 SW 1.2). Consequently the product model of each product level is different with the others. As well, in each level, various structure models may exist to express various business viewpoints. In addition to the transformations between the various business viewpoints models existing within each level, there exist other transformations between one level's model and another level's model. Then, while each product level has its specifications (attributes and models), consistency between product levels has to be maintained.

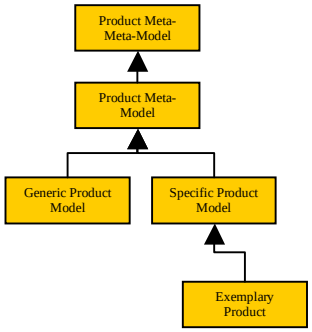


Fig 2, Different levels of Product model

A huge effort has been done during the last decade to study this function and adjust it to the special cases [Zina 06] [Carnduff 04] [Gzara 03]. However, all of the proposed approaches are about the first stage of PLM applications lifecycle, i.e. design stage. They don't consider later stages, once the PLM application is already deployed and has to evolve. Furthermore, the major part of these research works has been oriented to the first type of transformation of models, in which the problem is more concern to the interoperability between different business viewpoints. In other words, a lack of research on the evolution of currently-used PLM application is observed. The case of existing information system is different as the components of this system were not probably designed in such a way that the modification or transformation is predicted. The migration between the different levels of product model and the consequence of evolution in one level on others are the core of industrial case study.

Furthermore, Carnduff worked on the construction of a dynamic model that is able to store the information of evolution of a product during its lifecycle [These Art 022]. He presented a framework which captures the versioning the components of a product. With this framework, it is possible to trace the evolution of product model in each level, and facilitate the execution of model transformation which has been done on one level of product configuration to another.

3.2 Model transformation concept and approches

Model transformation framework is employed to facilitate the evolution of models, especially in the information science. It is used in various situations but the wide usage of MDA (Model Driven Architecture) approach makes the development of its methods and tools unavoidable. The MDA was created to help the design of a system by separating the specification of its operations from its platform in order to increase the portability, interoperability and reusability [OMG 03]. This approach improves the power of models in the process of design, implementation and modification of information systems. The modeling in MDA consists of three level of abstraction: level of model (level M1), the level of meta-model, which is a set of constructs and rules for constructing the model (level M2), and the level of meta-meta-model (level M3). Of course there is a layer of M0 that contains the instances of the models, in the other word, the data.

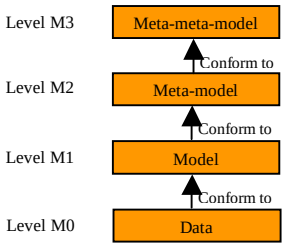


Figure 3, Levels of Abstraction in MDA

Moreover, the model-transformation methods are inevitable to support the reusability of models through information systems. These methods are employed to do mapping between different types of models, meta-models, platform independent models, platform specific models, etc. As well, there are lots of tools proposed to do this transformation.

In model transformation, the elements of target model should be specified based on the source model and this mapping is navigated by a set of rules. The semantics of the source model should be preserved in the target model. It means that in the transformation function, it should be envisaged that the *amount* of information stored in the source and target model should be verified and equalized.

There are some researches, mainly in information science, to categorize the different approaches or methods used to do the model transformation. Prakash divided the approaches on model transformation on three dimensions: the transformation techniques, the transformation language support and the genericity of transformation [Prakash 06]. Czarnecki also tried to present a classification of model transformation, with more the technical aspects, such as rules definition, relationship between source and target model and scheduling. Finally he classified the approaches on two main categories: *model to code* approaches and *model to model* approaches [Czarnecki 03]. Graph-based transformation, relational approaches which use the mathematical concepts, structure-driven approaches that concerns at first to set up the hierarchy of target model, direct-manipulation methods and the hybrid approaches are mentioned as the principal categories of the *model to model* transformation.

3.3 The originality of this paper.

As mentioned before, model transformation was the subject of many researches. But generally, it is rather studied in the context of information science, the programming and software industries with their own problematic, which is the reusability and interoperability of models. The objective of the current research is to employ it in order to realize the evolution of the product models. It requires more adoption of model transformation on the needs and specificities of PLM systems.

Furthermore, the researches done on model transformation in the context of information system design are more directed to the design of new system or model. In other words, the objective is how the new system should be designed in order to facilitate its evolution in the future. The evolution of a system which is designed formerly, without the forecast of this evolution is the subject of the current research. Therefore there is no

control and dominance neither on the source model and its meta-model, nor on the meta-model of target model.

4 Transformation Function

The proposed solution, as mentioned before, is to construct a transformation function which connects the semantics of source model to the semantics of target model, by defining a set of rules which also includes the constraints that govern the transformation. These constraints ensure the consistency between generated model and source model.

4.1 Meta model and Transformation function

As mentioned before, there are different incentives to evolve the PLM systems, especially its product model. With the perspective of product model evolution, these incentives may be classified as followed. Modification of product or its extension, within the enterprise, pushes to transform the product model, but *usually*, there is no need to evolve the product meta-model. Therefore the task is to migrate from one model to another with the remaining of meta-model unchanged.

In the other hand, changing the PLM system of the enterprise or in the case of unification and harmonization of PLM systems of different enterprises, usually it must evolve the product models, under the evolution of product meta-models. In PLM systems, different meta-models are used. Each system has its own meta-model structure which is not necessarily compatible with the meta-model of target system. For example, the meta-model used in Windchill is different compared with the Advitium. Therefore, somehow it is possible that this transformation function conform to a meta-method transformation function which does the transformation but in the M2 or M3 levels. The transformation function is also specified by a method. This method contains the specifications, rules and features required for the transformation. A meta-method of this transformation method is generic and all method transformation procedures conform imperatively to this meta-method.

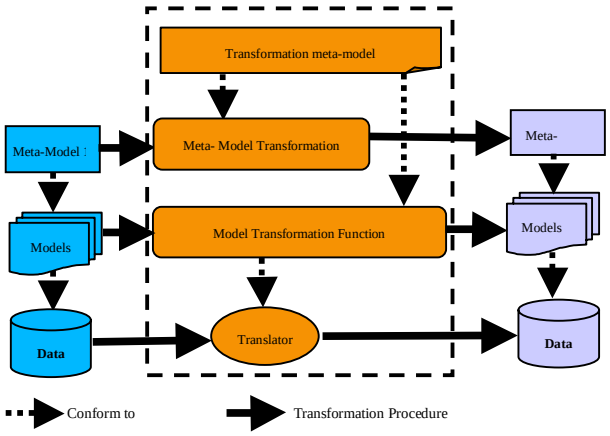


Figure 4, The model transformation structure

4.2 The migration of the data from a legacy configuration to new configuration

In the evolution of information systems, the migration of existing data, which have been stored as instances of the source models, to a format conform to the target models, is a crucial issue, since this is irreversible. Indeed, if the transformation has not been defined correctly and then the migration is done, the lost information can not been recuperated.

The transformation function constitutes a translator, which is written in a programming language and ensures data migration.

4.3 The industrial case study

Consider a company which produces tipper of truck. It uses its own information system, including PDM, but the regrouping with another company makes mandatory to adopt its PDM to the information system of the holding company. Consequently, it should also evolve its product configuration to the more general product configuration which is compatible with the products of other partners. In this context, the transformation method proposed in this research will be experimented with the industrial data. The case study is in the progress.

Another related problem of this enterprise is the modification of product configuration by the client. The client proposes a customized configuration compatible with its conditions. This configuration is studied in the company to evaluate its consistency with the tippers already designed. If it is well-matched with the company's product configuration, *generic product model*, the specific configuration will be derived. But if no, another *generic product model* should be designed. In the other hand, the product model at the "specifics" level can be evolved and modified by a client, in an *unpredictable* manner. Moreover it is interesting to study the influence of modification of product generic model on product specific model. It is desired to have a tool that perform the modification and evolve the other levels as a result.

5 Conclusion

The product model is a vital part of the PLM systems. This shows the importance of any modification or evolution of this part. In the current research, the model transformation system of MDA has been chosen as the framework and then the transformation function has been constructed, based on its meta-model. Analyze of the model transformation rules is inevitable.

Then a translator is designed to do the migration of existing data into the new format which is structured by the target models. In order to validate the results with a real situation, a company has chosen and the experimentation is under progress.

The product configuration evolution is a very immature subject. The companies and enterprises which are forced to modify their PLM, especially the product configuration, in order to adapt it to the new products, their partners, their suppliers or the subsidiary or mother company is growing. Moreover, this subject can help the designer of new PLM systems to present more compatible tools.

In the informatics side, the principles of model transformation and the methods and tools which are already designed to realize it, should be adapted to the case of product model evolution.

References

[Carnduf 04] Carnduf, T.W. (2004), `Configuration management in evolutionary engineering design using versioning and integrity constraints`, *Advances in Engineering Software*, Vol. 35, pp. 161-177

[Czarnecki 03] Czarnecki K. et al. (2003) `Classification of Model Transformation Approaches`, *OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture.*

[Eastman 99] Eastman Ch. Et al. (1999), `A database supporting evolutionary product model development for design`, *Automation in Construction*, Vol. 8, pp. 305-323.

[Eynard 06] Eynard B. et al. (2006) `PDM system implementation based on UML`, *Mathematics and Computers in Simulation*, Vol. 70, pp. 330-342.

[Eynard 04] Eynard B. et al. (2004) `UML based specifications of PDM product structure and workflow`, *Computers in Industry*, Vol.55, pp.301-316.

[Gzara 03] Gzara L. (2003)`Product information systems engineering:an approach for building product models by reuse of patterns`, *Robotics and Computer Integrated Manufacturing*, Vol.19, pp. 239-261.

[Krause 07] Krause, F.L. et al. (2007), `Meta-Modelling for Interoperability in Product Design`, *CIRP Annals - Manufacturing Technology*, Vol. 56, Issue 1, pp. 159-162

[Oh 01] Oh Y. et al., (2001) `Mapping product structures between CAD and PDM systems using UML`, *Computer-Aided Design*, Vol. 33, pp. 521-529

[OMG 03] OMG (2003), `MDA Guide Version 1.0.1`

[Prakash 06] Prakash N. et al. (2006) `The Classification Framework for Model Transformation`, *Journal of Computer Science*, Vol.2, No. 2, pp.166-170.

[Schuh 06] Schuh G. et al. (2006)`Product Structuring - the Core Discipline of Product Lifecycle Management`, *Proceedings of the LCE2006 - 13th CIRP International Conference on Life Cycle Engineering, Leuven (Belgium).*

[Svensson 02] Svensson D. (2002), `Strategies for Product Structure Management at Manufacturing Firms`, *Journal of Computing and Information Science in Engineering*, Vol. 2, Issue 1, pp. 50-58

[Sudarsan 05] Sudarsan R. et al. (2005) `A product information modeling framework for product lifecycle management`, *Computer-Aided Design*, Vol. 37, pp. 1399-1411.

[Zina 06] Zina S. et al. (2006) `Generic Modeling and Configuration Management in Product Lifecycle Management`, *International Journal of Computers, Communications & Control*, Vol. I, No. 4, pp. 126-138.