



HAL
open science

Model Driven Architecture to enhance interoperability between product applications

Néjib Moalla, Hanène Chettaoui, Yacine Ouzrout, Frédéric Noël, Abdelaziz Bouras

► **To cite this version:**

Néjib Moalla, Hanène Chettaoui, Yacine Ouzrout, Frédéric Noël, Abdelaziz Bouras. Model Driven Architecture to enhance interoperability between product applications. PLM'08, Jul 2008, Seoul, South Korea. pp.380-392. hal-00442252

HAL Id: hal-00442252

<https://hal.science/hal-00442252>

Submitted on 30 Oct 2018

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.

Model-Driven Architecture to enhance interoperability between product applications

Néjib MOALLA¹, Hanène CHETTAOUI²,
Yacine OUZROUT¹, Frédéric NOEL²,
Abdelaziz BOURAS¹

¹LIESP Laboratory, University of Lyon, CERRAL/IUT-Lumière
160 Boulevard de l'Université, 69500, Bron, France
Tel: +33 4 78 77 26 27; Fax: +33 4 78 00 63 28
firstname.lastname@univ-lyon2.fr

²Laboratory G-SCOP, INPGrenoble – UJF - CNRS
BP53, 38041 Grenoble Cedex, France
Tel: +33 4 76 82 51 29; Fax: +33 4 76 82 70 43
firstname.lastname@g-scop.inpg.fr

Abstract: Throughout product lifecycle, the definition of product data is confronted to the diversity of the business activities and influenced by their specific information systems. This paper is related to the analysis of technical and functional product data exchange solutions throughout product lifecycle. In such context, the communication between partners is ensured by setting agreements, methodologies, and adopting standards. However, these communication issues have to deal with systems heterogeneity.

We will mainly focus on two components of the lifecycle: Design and Supply chain. Design teams must collaborate effectively with their internal and external partners to develop competitive products. Product data must be reviewed according to the product specifications throughout its supply chain to ensure the quality of product data definition. In both cases, the interoperability between computer systems and applications presents a vital issue and an important challenge to address.

Today, the OMG's model-driven architecture (MDA) makes available an open approach to write specifications and develop applications, separating the business and application functionalities from the platform technology. We will focus on the use of MDA technologies to improve the interoperability and illustrate its application through two case studies related to the Design and Supply Chains.

The first case study shows how to enhance the interoperability between design experts exchanging heterogeneous CAD/CAM models and using the formulation of business knowledge throughout a Model-Driven Architecture approach. The second case study illustrates the contribution of a model-driven interoperability approach to ensure data quality in the logistic perimeter of the vaccine supply chain.

Keyword: Product Data Exchange, Model-Driven Architectures, Model-Driven Interoperability, Collaborative Product Design, Product Data Quality, Design Chain, Supply Chain.

1 Introduction

The complexity of business processes and the need to enhance a collaborative product development present some permanent inputs for the development of interoperability concepts. In fact, interoperability is widely considered as a key factor for time-to-market and cost reduction. The need to communicate and exchange product data more rapidly is faced to different types of limitations related to the specificity of each context in terms of business activities, specific product knowledge's, defined processes, deployed application and data structuring formats. The interoperability concepts are developed to bring some adapted solutions for industry in its permanent evolution. The evolution of the Model-Driven Engineering theory gives several attractive solutions with some standardized architectures that seems to be useful to extend the interoperability concepts.

We explain in the second section of this paper the need as well as the contribution of interoperability concepts in connecting software application in the design and supply chain contexts. In the third section, we propose a generic approach to ensure interoperability between information systems based on model-driven concepts. We explain in the fourth section two cases studies and we provide after some conclusions and perspectives.

2 The need for interoperability

Throughout product lifecycle, many information systems are deployed to support product and process definitions. Each business activity is covered by some specific information systems. Consequently, this diversity is more perceptible in data structuring formats and languages and still voluntary supported by information systems providers to preserve their identity and enhance their competitively in a market share context. With the huge number of proposed solutions and the need to enhance the communication between them, some gaps are identified and some specific efforts are needed to connect these systems. The interoperability concepts provide some solutions in this perspective.

The concepts of interoperability are widely defined in the literature (IEEE, 1990) as "the ability of two (or several) systems or components to exchange information and thus to use exchanged information". The ISO 16100 (2000) part 1 defines the interoperability of manufacturing systems as "the ability to share and to exchange information using a common syntax and semantic to reach a specific functional relation through the use of a common interface". [1, 2] give more information about the specificities of interoperability concepts.

The European project ATHENA [3] and INTEROP [4] identify five abstraction levels for interoperability between systems: the business level, the knowledge level, the process level, the application level, and the data level. Each abstraction level presents some specific models. Interoperability is ensured when all these models are well connected. The following figure (figure 1) illustrates the structuring of these abstraction levels and the research fields to develop them.

The interoperability concepts present a vital issue for a collaborative design approach. In fact, the design of a complex product is generally done by more than one designer,

Model-Driven Architectures to enhance interoperability between product applications

geographically dispersed and using different software solutions according to their expertise. However, the collaborative design is faced to the heterogeneity of systems and product data structuring inside. In a collaborative Computer Aided Design (CAD) context, we can identify three ways to communicate product information between heterogeneous CAD tools: (i) the use of integrated CAD software modules from vendors; (ii) the use of standards for data exchange such as “Initial Graphics Exchange Specification” (IGES) or the Standard for the Exchange of Product model data (STEP) and finally (iii) the use of some specific proprietary data exchange formats. This communication problem between CAD systems is translated when communicating product data from CAD to Computer Aided Manufacturing (CAM) Systems. Product Data management (PDM) is then considered as the main system to share files issued from CAD/CAM systems providing a correspondence between files describing the same artifact. PDM reflects the lack of interoperability between technical formalisms.

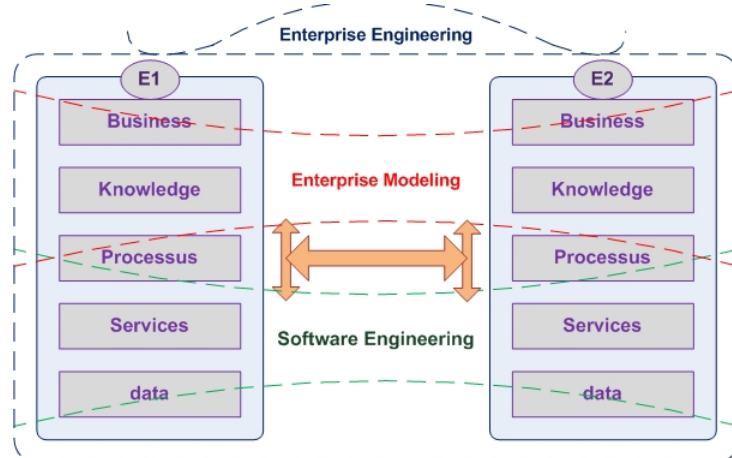


Figure 1: Interoperability abstraction levels [5]

Furthermore, in the supply chain perimeter, the Enterprise Resource Planning (ERP) is considered as the main system supporting product definition as well as some business activities. The ERP system interacts with many information systems in this perimeter and develops a bidirectional interaction. When business opportunities expand, the product definition evolves to be aligned with the new defined strategic target. The preserving of product data quality definition in the ERP system becomes a big challenge when the interaction between information systems is performed according to different business positions and priorities. The concepts of interoperability between heterogeneous information systems are needed to be deployed in the supply chain perimeter.

Whether in the design stage or in the supply chain stage of product lifecycle, there is a common need to enhance data exchange between different deployed solutions. Since each system evolves in a specific business context and presents its specific models, we propose in this work to focus on the Model-Driven Engineering (MDE) concepts to propose some adapted interoperability solutions.

3 The proposed approach

The Model-Driven Engineering (MDE) allows defining new enterprise software applications using existing models by refining them until the generation of code. The key concepts of MDE are the modeling and the model transformation activities. The Model-Driven Architectures (MDA) are the standardized part of MDE concepts defined by the OMG. MDA are considered as an incremental and iterative software development process, centered in the architecture, driven by models transformations and business process reuse and based on some frameworks and technical components [6].

The Model-Driven Architecture (MDA) [7] approach marks the new generation of software engineering based on models technologies. In fact, when we need to develop a new software application to be integrated in a specific perimeter, it is possible to take advantages of existing models already optimized for this context to define targeted software application [6]. The MDA approach is identified to be the least expensive and the fastest method to build new applications [8]. Developed by the OMG, the MDA approach consists in defining four abstraction levels [9]:

- The Computational Independent Model (CIM) level is a software independent model used to describe a business system.
- The Platform Independent Model (PIM) level, as a transformation of the CIM, consists of a model with a high level of abstraction that is independent of any implementation technology. Within a PIM, the system is modeled from the viewpoint of how it best supports the business.
- The Platform Specific Model (PSM) level, as a transformation of the PIM, is tailored to specify the system in terms of the implementation constructs that are available in one specific implementation technology.
- The final step in the development is the transformation of each PSM into code. Because a PSM fits its technology rather closely, this transformation is relatively straightforward.

From one abstraction level to another, the MDA Framework structures the transformation process and ensures the mapping of metamodels expressed in MOF as the metamodel language defined by the OMG [10]. The figure 2 presents an example of application of the MDA framework to transform models from the PIM level to the PSM Level.

Model-Driven Architectures to enhance interoperability between product applications

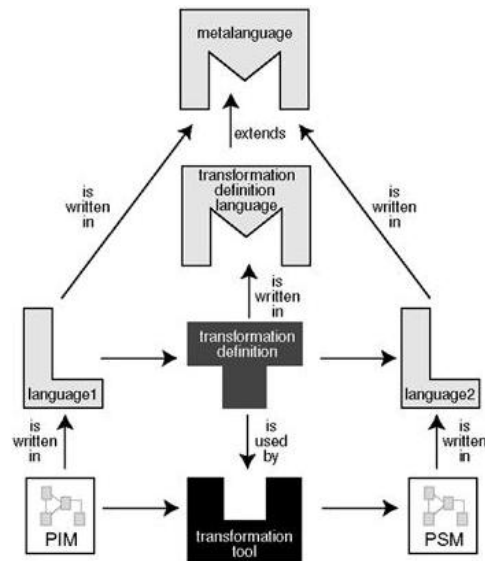


Figure 2: MDA Framework [11]

In the perspective to ensure interoperability between existing applications, the MDA approach can be applied through the definition of the Model-Driven Interoperability concepts. In fact, according to each interoperability abstraction level, each system presents its specific models. To interoperate two heterogeneous systems it is possible to connect their models defined at each interoperability abstraction level through the different MDA abstraction levels. The model transformation process is used in two issues:

- The first issue aims at transforming a defined model from one MDA abstraction level to another. It is a vertical transformation that will be improved when generated models are merged with those coming from the corresponding interoperability abstraction level (figure 3)

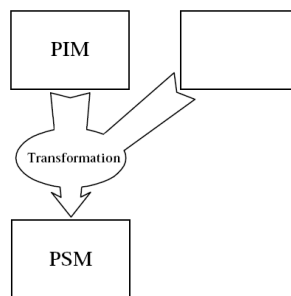


Figure 3: Vertical model transformation [7]

- The second issue aims at transforming models in the same MDA abstraction level in order to refine them or for example to change their modeling language. It is a horizontal transformation such as from PIM to PIM model (Figure 4).

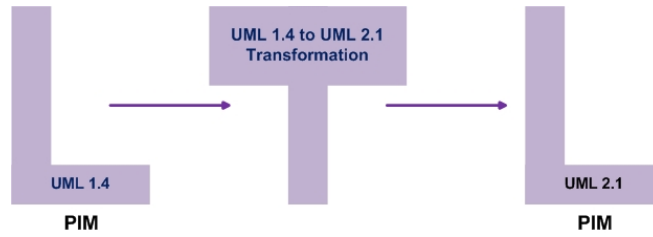


Figure 4: Horizontal model transformation

We illustrate in the following figure (figure 5) the correspondence of the MDA abstractions levels with the four modeling layers defined by the OMG. All the models defined at the interoperability abstraction levels feed the MDA modeling levels with models issued from heterogeneous systems implied into the interoperation.

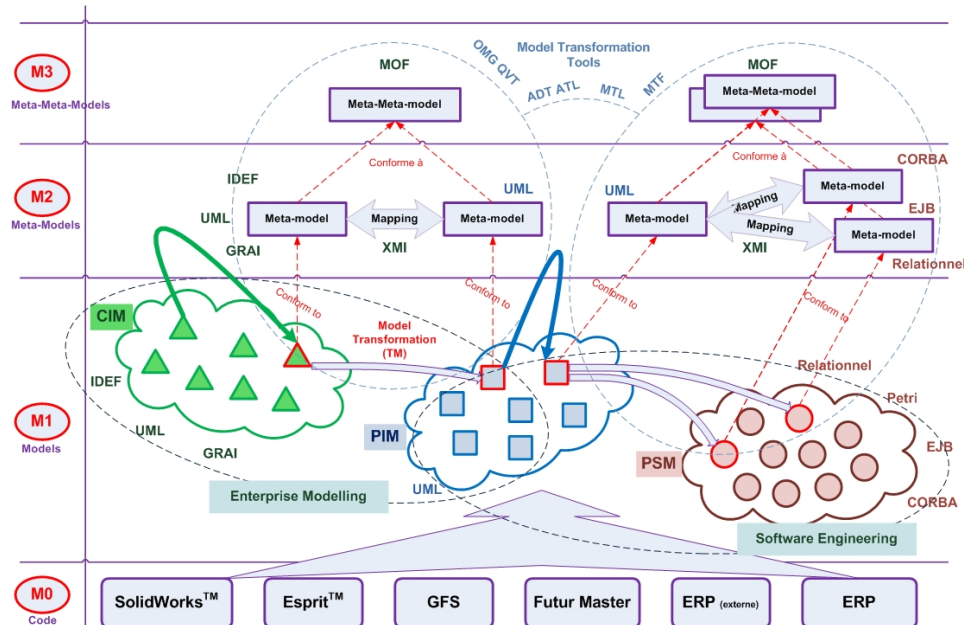


Figure 5: Model-Driven interoperability architecture [5]

We present in the following section two cases studies about the application of the MDA concepts to ensure the interoperability between some heterogeneous information systems.

4 Cases studies

Two cases studies are presented in this paper: the first is dedicated to the design chain context and deals with product data exchange between CAD/CAM systems. The second is dedicated to supply chain context and deals with product data quality in the Enterprise Resources Planning ERP system.

Model-Driven Architectures to enhance interoperability between product applications

4.1 Case 1: interoperability between CAD/CAM tools

4.1.1 Collaborative design use case

Design and engineering of systems is increasingly becoming distributed and collaborative [12]. Combining multiple tools in a design process is a natural way to design a complex product. The effort for translation from CAD system to CAM system is costly. In a design process, CAD and CAM tools are applied to design and to manufacture product. Unfortunately, tools remain separated and interoperability is not formalized

The following scenario illustrates interoperability issues between CAD and CAM tools. Then we present an improvement of technologies based on Model Driven Architecture approach.

Two modeling formalisms are concerned by this scenario:

- Within a CAD system (we used SolidWorksSM), designer is in charge to model the geometry of the product. The designer knows how to take into consideration client's requirements and convert them into geometric constraints.
- Within a manufacturing system (we used EspritSM), manufacturer is in charge to prepare the manufacturing operations of products.

Usually, the designer models a new product with the requirements and functionality fulfilling the market demands. Then he transfers his model to the manufacturer exporting it into a standard format. With his tool, EspritSM, the manufacturer prepares manufacturing operations and suggests some minor modifications to original design to take into account his own expertise. For example a requirement for the diameter of the drilling tool to respect the available tools and an extrusion length to minimize material to be removed. Until now, there is no support to navigate this kind of information back to designer. Currently, data exchange is carried out by standards. Exchanged data are static, one-way; they do not incorporate details such as sketches, constraints and features, which is a first representation of the designer's intent. Some recent works attach manufacturing information to the CAD model. These work are developed under the STEP-NC approach [13]. STEP-NC files are generated for the CNC machines. This concept does not fulfill the interoperability requirements. Interoperability is seen as a key asset in future.

4.1.2 An environment to assist synchronization between heterogeneous models

To validate MDA capacity we use the GAM framework [14]. GAM is a project developed to provide an experimental platform for collaborative design in product development processes. It provides tools to manage information used along the product life cycle. Other works are already provided to define integrated meta-models with a similar goal as GAM. The intention in GAM is to build a very generic modeling framework enabling the easy management of every candidate meta-model. The framework could be used to compare and to modify them with respect to the usage point of view.

In GAM, we instantiated the PPO Meta-Model as a middleware for exchanging product information. PPO (Product Process Organization) Meta-model is the result of integration of three models: a product meta-model, a process meta-model and an organization meta-model [15]. It is a light meta-model (with few concepts) but evolutionary.

The PPO environment is expected to be the middleware for data exchange [16]. Designers and manufacturers should be able to interact with this environment without any costly effort. PPO collaborative environment cannot integrate specific rules to the

processing of data resulting from Esprit™, else it should be done for any other expertise tool. It is thus chosen to let each expertise define connections rules to PPO framework.

To interoperate SolidWorks™ and Esprit™ we use a MDA based approach. Our methodology consist in bring into the PPO environment each heterogeneous model. The goal is to get a common shared model. Synchronizing homogenous PPO models is the needed step to satisfy this goal.

A first step is to extract the business model from business tools. This step should be carried out by the business software developer. Figure 6 presents the SolidWorks™ model extracted automatically from the SolidWorks™ tools in the GAM format and conforming to the SolidWorks™ meta-model already defined by the SolidWorks™ developer. At the same way, Figure 6 presents the Esprit™ model extracted automatically from the Esprit™ tools in the GAM format and conforming to the Esprit™ meta-model already defined by the Esprit™ developer.

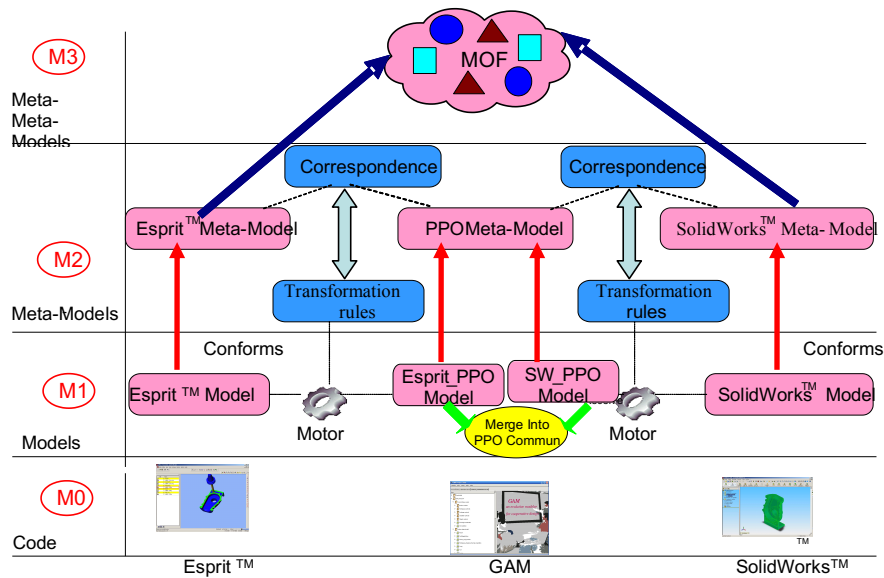


Figure 6: Business Model Interoperability

An automatic extraction is provided by an API which learns automatically all the required parameters. Required parameter should be specified by the expert tool. Business experts should describe its business meta-model in the GAM environment. This meta-model allows the creation of a library that joined to the business tool library enable the creation of the GAM model stating from the business model. The created API checks if GAM model is already created or not. If created model required modification are achieved, if not, required model is generated: we follow this verification via a versioning mechanism included in GAM.

Once we have the Esprit™ model and meta-model, we convert it to a PPO model according to converting rules meta-model specifying correspondences between business model and PPO model. Last step is to synchronize homogenous model (Esprit-PPO-

Model-Driven Architectures to enhance interoperability between product applications

Model and SW-PPO-Model). This step needs negotiation to converge to a common model. The PPO model provides decomposition and multi-views ability to build the common model at an easy way.

Now when a diameter is modified in Esprit™ we can propagate this information towards the common PPO model, to get the corresponding parameter in SolidWorks™. There is only the need to negotiate whether to keep the expert suggested value or not. In this case (as presented in Figure 7) the manufacturer working with the CAM tool suggests an “angle” of “0.2” to achieve the machining operation with his available fabrication tools. The designer working with the CAD tool agrees with the modification, a PPO synchronized model is created and each expert reads this new value thanks to the read/write API that we create. Here we see how models and meta-models could express expert’s knowledge and how expert participate actively to define each step of the process. Our environment provides an assistance to enable the expert to express collaboration need’s stating from the extraction of the expert business model, the specification of the connection rules, and the synchronization of homogenous models.

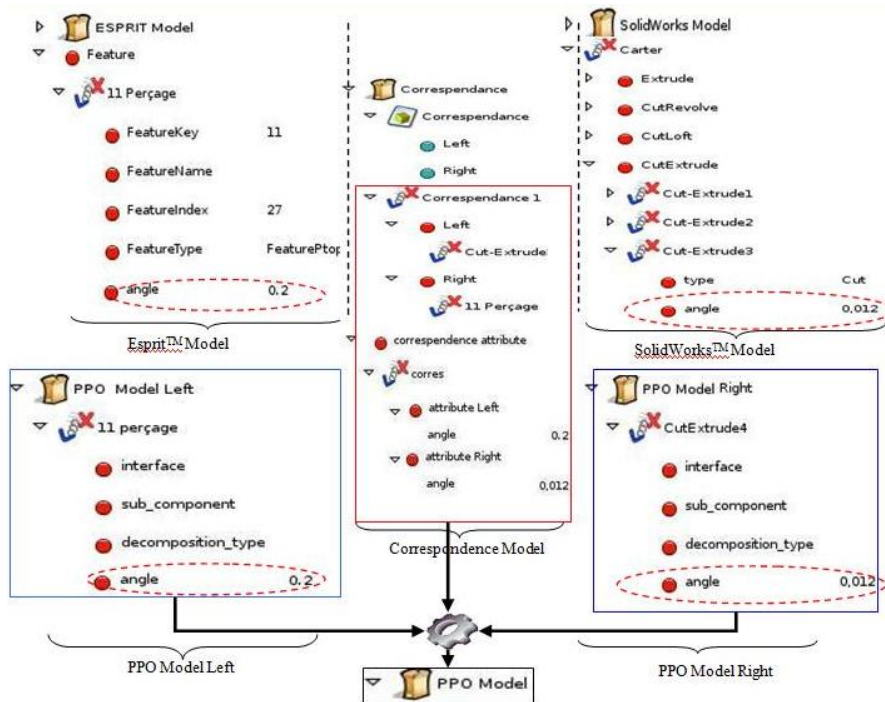


Figure 7: Correspondences at the model level between Esprit™, PPO and SolidWorks

In this example the significant parameters to share is the attribute “angle”, the value specified by the designers does not fulfill the manufacturer demands and machining tools availability. The Correspondence presented here is at the “Class” level. We defined a correspondence at the “attribute” level. Here the Feature “Cut-Extrude3” issued from the CAD model corresponds to the “11 perçage” issued from the CAM model.

Once the PPO synchronized model is generated, each expert’s tool read the modified value and update his business model. Experts continue to work in their models and do

modification. A first step towards interoperability is reached at a finer level than the connected files in a PDM.

4.2 case 2 product data quality

This case study deals with the deployment of a MDA approach to ensure interoperability between information systems in the supply chain perimeter of vaccine industry. It is developed within Sanofi Pasteur France company in order to ensure product data quality.

4.2.1 The vaccine supply chain

Due to the complexity of vaccine product, the main production information system is an Enterprise Resource Planning (ERP). In the supply chain perimeter, the ERP system supports some business modules and interacts with other systems from different business activities. Product data quality is rapidly altered when these systems evolve independently.

To ensure data quality, we choose to target a reduced number of product data, identified as the most shared and the mainly critical in the vaccine supply chain perimeter. As examples of these data we identify, the *manufacturing site* of one component in a given product, the *shelf life*, the *storage conditions*, the *batch size*, etc. These data are identified as critical because they present an important impact in regulations respect, planning stability, production processes, etc. Each data is impacted by the evolution of one or many information systems in our perimeter. When these systems evaluate, it is very difficult to maintain the quality of critical product data. Therefore, we propose in the next step to analyze the specificities of each critical product.

4.2.2 The application of MDA concepts

Our approach aims at its perspective to develop a new software application in order to manage product data lifecycle. The deployment of the MDA Framework [17] allows to integrate all business models, already defined and optimized in different business contexts, in one Computer Independent Model (CIM). This model is transformed to a functional and architectural model at the Platform Independent Model (PIM) level. Through the transformation process, the generated Platform Specific Model (PSM) is intended to be transformed to obtain the code of a new application. However, we propose in our particular context to make use of this PSM when it is expressed in the same platform than the information system where we need to ensure data quality. Data quality is achieved when we compare new generated model with the exiting one. In the mapping process, we aim to feed existent application models with all entities needed to maintain data quality under a global vision of product data lifecycle. We present in figure 8 an overview of our architecture when applied to ensure data quality in the Enterprise Resource Planning (ERP) system.

Model-Driven Architectures to enhance interoperability between product applications

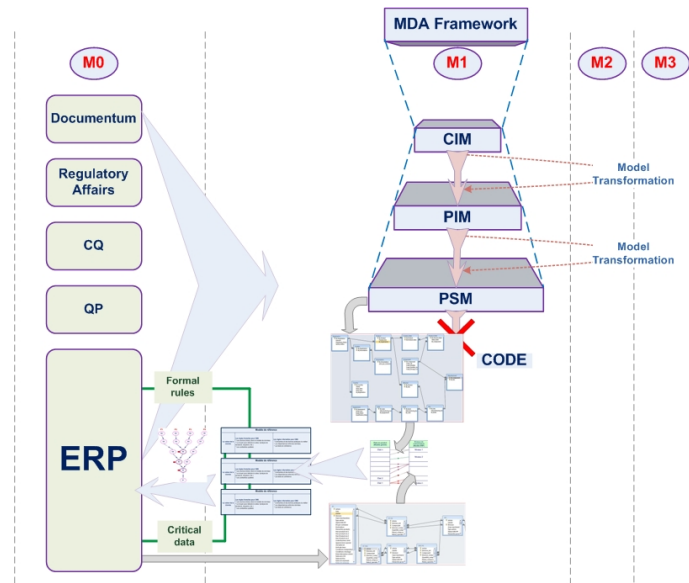


Figure 8: The application of MDA concepts

In the next step, we explain the data quality process using the new PSM model to ensure data quality in the ERP system.

4.2.3 Data coherence analysis

When we deal with data quality, we address some quality dimensions to analyze and enhance. To ensure the quality of critical product data in the ERP system, we choose to address following data quality dimensions:

- The accuracy of the data value.
- The validity of the data value in the ERP system.
- Conformity of data value to its definition source as well as to its definition in the vaccine supply chain perimeter.
- The coherence of data value according to the production process.

To reach these quality dimensions, we propose to define a Reference Model as a framework to structure:

- The product data value to define.
- Some formal rules to structure all rules needed to update a product data value in the ERP system.
- Some informal rules defined from the mapping of the new PSM model generated through the MDA framework and the one from the ERP. As example of informal rules, we can identify those related to the mapping process as correspondence, aggregation, of interpretation; those related to the semantic correspondence between data values, etc.

Using these defined informal rules, product data value is defined for all product components independently. To be conforming to the definition and the structuring of product data in the ERP system, several coherence rules need to be defined. In fact some components enter in the composition of several products (figure 9). The definition of a common component data value in the ERP must be coherent with the structuring of the entire product and especially with the specifications of product destination country.

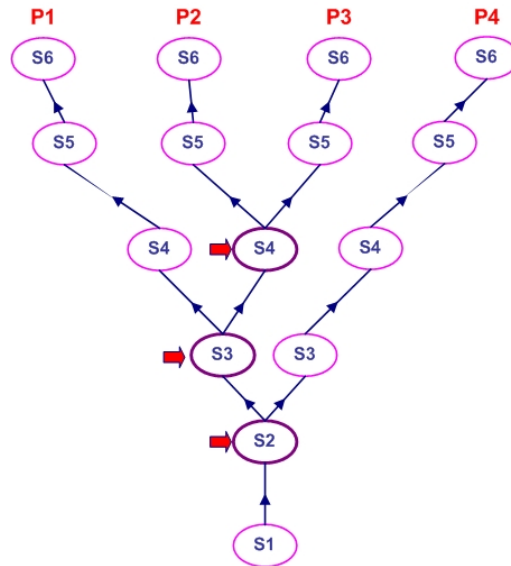


Figure 9: Component reuse

As example, the component S3 enters in the composition of the three products: P1, P2 and P3.

5 Conclusion

In this paper, we addressed the need for interoperability to enhance product data exchange between heterogeneous information systems. We explore the concepts of the Model-Driven Architecture (MDA) and we propose two cases studies to explain the deployment of such approach in a collaborative design context, as well as in a supply chain context.

Our contribution in the first case study consists on: (i) the creation of facilities to plug a new representation and extract business model in the GAM format; (ii) the assistance for the creation of correspondence between business model and the collaborative model; and finally; (iii) the management of interaction between commercial tool used by industry and the proposition of a strategy to connect several models together.

Our contribution in the second case study consists on: (i) the integration of diverse business constraints when validating product data quality; (ii) the generation of a common product data model to share in the supply chain perimeter; (iii) and the expansion of the trust in the quality of the ERP data in its perimeter by providing a common validated product data values.

With the forthcoming steps, we aim to continue the deployment of proposed concepts within different business contexts in industry.

Model-Driven Architectures to enhance interoperability between product applications

Acknowledgment

This work is elaborated under the regional research project AncarPLM developed in collaboration between LIESP Laboratory from the University of Lyon and G-SCOP Laboratory from the University of Grenoble.

References

1. Chen, D.: Enterprise Interoperability: Basic Concepts, Definitions and Approaches. Tutorial in Interop-ESA 2007, Madeira, Portugal (2007),
2. INTEROP-NoE: Practices, principles and patterns for interoperability. NoE IST-508 011 (2005) 307,
3. ATHENA-IP: Second Version of State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability. In: WD.A1.1.2, W.D. (ed.). ATHENA Project No 507849 (2005) 212,
4. INTEROP-NoE: State-of-the art for Interoperability architecture approaches. INTEROP-NoE Project (2004) 362,
5. MOALLA, N.: Product Data Quality Improvement in The Vaccine Lifecycle Context: a Model-Driven Interoperability approach. University of Lyon 2, Vol. PhD. University of Lyon 2, Lyon (2007) 183,
6. Kadima, H.: MDA : Conception orientée objet guidée par les modèles. DUNOD (2005)
7. Miller, J., Mukerji, J.: MDA Guide. In: OMG (ed.). OMG (2003) 62,
8. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer (2007)
9. Blane, X.: MDA en action. Eyrolles (2005)
10. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture™: Practice and Promise, Vol. 1. Addison Wesley (2003)
11. Kleppe, A., Warmer, J., Bast, W. (eds.): MDA Explained: The Model Driven Architecture™: Practice and Promise, Vol. 1. Addison Wesley (2003)
12. Danesi, F., Gardan, N., Gardan, Y.: Collaborative design: from concept to application. Proceedings of the Geometric Modeling and Imaging. IEEE, London UK (2006),
13. Sääski, J., Salonen, T., Paro, J.: Integration of CAD, CAM and NC with Step-NC. VTT Industrial Systems (2005), <http://www.vtt.fi/inf/pdf/>
14. GAM: GAM Project. Vol. 2007, Grenoble (2007), <http://www.g-scop.inpg.fr/GAM/>
15. Noël, F., Brissaud, D., Tichkiewitch, S.: Integrative design environment to improve collaboration between various experts. Annals of the CIRP 52 (2003) 109-112
16. IPPOP, Vol. 2007 (2007), <http://projects.opencascade.org/IPPOP>
17. Mellor, S.J., Scott, K., Uhl, A., Weise, D.: MDA Distilled: Principles of Model-Driven Architecture. Addison Wesley (2004)