International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:8, 2014

Transformation Method CIM to PIM: From Business Processes Models Defined in BPMN to Use Case and Class Models Defined in UML

Y. Rhazali, Y. Hadi, A. Mouloudi

***Abstract*—**This paper proposes a method to automatic transformation of CIM level to PIM level respecting the MDA approach. Our proposal is based on creating a good CIM level through well-defined rules allowing as achieving rich models that contain relevant information to facilitate the task of the transformation to the PIM level. We define, thereafter, an appropriate PIM level through various UML diagram. Next, we propose set rules to move from CIM to PIM. Our method follows the MDA approach by considering the business dimension in the CIM level through the use BPMN, standard modeling business of OMG, and the use of UML in PIM advocated by MDA in this level.

***Keywords*—**Model transformation, MDA, CIM, PIM.

I. Introduction

MDE (Model Driven Engineering) [4] is an alternative approach which aims at the development of information systems, based on the creation of source models and transforming them to multiple levels of abstraction until we automatically get a code. Its objective is to automate the process of software development that the specialists follow manually. MDE is a general approach that can be seen as a family of approaches, where MDA (Model Driven Architecture) [1] supported by OMG, is presented as the most interesting and the most common variant. MDA has the same principle as MDE, but it provides its own characteristics defined in three levels of abstraction, defines some requirements to be respected, and also recommends the use of some standards.

The first level of MDA is the CIM (Computation Independent Model) presented as models used by business managers and business analysts to describe the business process. The second level is the PIM (Platform Independent Model) which allows defining the models used by analysts and the software designers to realize an independent analysis and a conception of the developed software. The third level is the PSM (Platform Specific Model) which is considered as models of code used by software developers. These models are believed to contain all the information needed to operating an execution platform and used by software developers. The code is not a model of MDA, but it is the final result of the MDA process.

Y. Rhazali and A. Mouloudi are with the ACIDROS Laboratory, Faculty of science, Kenitra, Morocco (e-mail: rhazali.yassine@hotmail.fr, mouloudi_aziz@hotmail.com).

Y. Hadi is with the LARIT Laboratory, Faculty of science, Kenitra, Morocco (e-mail: hadiyoussef@gmail.com).

Transformations between the different levels of MDA begin with the transformations from CIM to PIM that aim to partially build, PIM models from CIM models. The goal is to rewrite the information existed in the CIM models into PIM models. These transformations are going to ensure that business information is conveyed and respected throughout the MDA process. Then, the transformation of PIM models to PSM models adds PIM technical information related to a target platform.

In practice, the automatic transformation starts from the second PIM level. However, our ultimate goal is to make the CIM level a productive one, and also a basis for building PIM level through an automatic processing so that the business models would not only be documents of communication between business experts and the software designers.

In this paper, we propose a solution to induce the automation of the transformation of CIM level to the PIM level by studying how to use the BPMN, standard of business modeling, effectively so as to achieve focused CIM models to simplify the transformation to PIM. Then, we define a set of rules to automate the transformation to a PIM level.

Our approach uses the BPMN [3] collaboration diagram and process diagram which represent standard business model to present the CIM level. Then the rich business models which contain concentrated information help us to achieve UML [2] models of PIM level. The first model of the PIM level is the use case diagram that defines the functionality of the information system, however, the model class diagram models the system classes and their relationships independently of a particular programming language.

The rest of this paper is organized as follows. Section II analyzes the related work of the CIM transformation to PIM. Section III presents our approach and describes the rules for constructing models of CIM level and the rules for transformation from the CIM level to the PIM level. In Section IV we illustrate our proposal in a case study showing the construction of the CIM level and the transformation to the PIM level. Finally, in Section V, we conclude by determining the outcome of our work and describing future works.

II. Related Work

In this section, we are also going to shed light on the related work concerning the passage of the CIM level to the PIM level in MDA drawing in part the advantages and disadvantages of each approach.

The oriented service transformation from CIM to PIM was

proposed by Castro et al. [6]. The authors present the CIM level using BPMN for modeling business process and the value model [5] so as to identify services from the beginning in the business perspective. Through the ATL language, the authors move towards a PIM level presented by two extensions of use case model and two extensions of the activity diagram. Although this method has the advantage of identifying services and the specification of a business process at the CIM level in order to guide the transformation to PIM in a semi-automatically manner with well-defined rules, but the authors are only limited on the use case diagram and the activity diagram in the PIM level and do not present the structural view (generally through the class diagram) that defines the ultimate objective of this level. Also, the use of activity diagram in the PIM level is causing great inconvenience since this diagram is among the standards for modeling business processes.

A transformation approach from CIM to PIM based on security requirements from the beginning in the business perspective is presented by Rodríguez et al. [7]. The authors use the BPMN notation for modeling business processes secure of the CIM level, then, they determine the transformations in QVT [12] in order to obtain class diagrams and use case. This method manifests a reference in the transformation of CIM to PIM security oriented. However, this proposal focuses only on secure information systems.

Hahn et al. [9] focus on engineering services driven by models. The authors present the CIM level with BPMN notation and establish the ATL language to achieve a transformation to the PIM level represented in this approach by using SoaML models [15]. The authors use SoaML, the new OMG standard for modeling services, but this approach does not have the ultimate goal of PIM level which is reflected in one or more structural diagrams.

Zhang et al. [10] describe an approach in which the CIM and PIM are respectively represented by features and components. Responsibilities in this demarche are considered as connectors between features and components to simplify the task of transformation CIM to PIM. Grammel and Kastenholz [11] rely on a DSL connection, which focuses on the management of traceability in general. Both approaches offer solutions to transform CIM to PIM, while they do not specify models used in CIM level and PIM level.

An approach, respecting MDA, aims at transforming the diagram use case to the activity diagram is proposed by Gutiérrez et al. [13]. The authors use QVT to transform existing use cases to the activity diagram. While this approach makes a CIM to PIM transformation through clear rules, but the authors define in the CIM level functional requirements represented by the use case.

Mazon et al. [14] propose an objective-oriented approach by defining a UML profile to present the CIM level, based on the i* modeling framework. The authors use QVT to move towards the PIM that focuses on conceptual modeling of data warehouse. However, this approach only tackles the transformation in the field of data warehousing.

Kherraf et al. [8] propose a disciplined approach to transform the CIM to PIM using the business process model and use case diagram as an initial step in the modeling of business processes and a detailed activity diagram which defines the requirements system represents the last step in the CIM level. The elements of the requirements model are transformed as components of the system. These are presented in the component diagram as a first step in the PIM level. Finally, a set of business archetypes helps to transform the system components to obtain the class diagram. This approach offers interesting ideas on transforming the CIM to PIM. However, this approach uses diagram use case that represents the system functionalities in the CIM level.

Kardos et al. [16] present an analytical method for the transformation of CIM to PIM in MDA. The authors define the CIM level through the data flow diagram, then use the use case diagram to initiate the information system view. This approach also defines a model of activity diagram as well as a model of sequence diagram and finally a model class diagram. The advantage of this method is the use of various UML diagrams that present different views of the information system in PIM level, but this method does not present a real business view since it uses DFD, and does not clearly define rules for transforming the CIM to PIM.

After this overview on related work concerning the passage of the CIM level to the PIM level, we can classify the work into five categories. Research works that use requirements models (such as the use case diagram) in the CIM level, to facilitate the transformation to PIM [13], [8]. Then, other research works [6], [9] even if they define the business processes in CIM level, do not represent the structural view (usually through the class diagram) in the PIM level. Then there are researches that are moving towards transformation in a particular field [7], [14]. There are also methods such as [16] which represent the structural view in the PIM level and do not oriented to a particular area, but the authors do not specify transformation rules. Finally, there are methods [10], [11] that just define the transformation rules and do not have the models used in the CIM and PIM level.

So we can say that the main contributions of our study compared with others: a business process model is used in CIM level, then, we define the structural view in PIM level, next we propose an approach of generic transformation that is not directed to a particular field and has clear rules to achieve the maximum possible automatic transformation from CIM to PIM.

III. Proposed Method of Transformation from CIM to PIM

A business model is the abstraction of how the business is working. According to our modeling purposes we can, more or less, make different models to describe the same reality. Also, we can model so as to master business process, enhance the process of communication with customers or partners, or to make an information system. In this paper, our goal is to design a business process models as a first step in the process of developing an information system. In this case, we are faced with two possibilities: make business models in the form

of documents to be transformed manually via analysts and software designers, or made business models to be transformed in an automatic way. Given that we are relying with the MDA approach, the second choice was considered in our approach in order to design effective business models that contain useful information to facilitate the transformation to a low level of abstraction.

Our proposal considers the business dimension in the CIM presentation level through the use of real high-level business models to preserve the business knowledge during the transformation to the PIM level in order to achieve a quality information system.

In this approach, we use the BPMN collaboration diagram and process diagram, standard modeling business processes, in order to take advantage of each diagram to achieve a rich and concentrated CIM level, which simplifies our transformation to the PIM level.

MDA recommends the use of UML in the PIM level, we present this level with model use case diagram which shows the functional point of view and model class diagram who describes the static view of the information system.

The models of the PIM level are obtained through an automatic transformation of CIM level, via well-defined and concentrated transformation rules.

Below, we present the rules of construction of CIM level and the rules of transformation to the PIM level.

A. Rules of Construction CIM Level

1. The Rules for Constructing the Model of Collaboration Diagram BPMN:

- Define means and not complex sub-processes, i.e., must not contain other sub-processes. In fact, each sub-process must be comprised of about 4 to 12 tasks.
- If a sub-process consists of less than 4 tasks, or represents a complementary operation to another sub-process, we merge two sub-processes into one.
- Avoid the maximum possible, the representation of tasks and present only manual tasks.
- The model does not describe all possible cases and paths, but it just presents a description of the sequence of activities of the most common business processes.
- Focus on sub-processes and their sequences.
- Identify the maximum possible of the actors who interact and who collaborate in the achievement of business processes since we are talking about an enterprise process.
- Avoid in this level, the maximum possible, representation of the gateways.

2. The Rules for Construction the Model of Process Diagram BPMN:

- Detail individually each sub-process in a model as several tasks.
- Do not represent the manual tasks of model collaboration diagram.
- Represent the gateways in this model.
- Enrich this model with the most exceptional ways.
- Add a data object at the output of each task.

B. Transformation Rules from CIM to PIM

3. The Rules of Passage from the CIM Level Models to Model of Use Case Diagram:

- Every task of the model process diagram that corresponds to a functionality of the system is transformed to use case.
- The collaborator, who realizes the sub-processes of the model of collaboration diagram BPMN, becomes an actor use cases that correspond to the tasks of this sub-process.
- If there is "exclusive gateway" between two tasks, the corresponding use cases connect by a relationship "*extend*".
- If there was just a sequence flow between two tasks, the corresponding use cases connect by a relationship "*include*".
- Do not transform the sequence flow returning back.
- Each sub-process of collaboration diagram model is transformed to a package which includes the use cases corresponding to the tasks of this sub-process.

4. The Rules of Passage from the Model of Process Diagram to Model of Class Diagram:

- Transform data objects of model process diagram as classes.
- If several data objects that have the same name, they are transformed to one class.

IV. Case Study

In this section, we present a case study for sales through e-commerce to illustrate our approach of transforming the CIM level to the PIM level.

A customer can browse the catalog of products available, he can also see detailed information about each item, then he decides to put a quantity of product in the cart or not. Each time the customer has the right to change the amount or eliminate completely the article from the cart. Once products that satisfied the needs of the customer are clearly defined, the latter starts the command, then, he represents the payment information, and precise details of delivery.

An order agent begins treating the order, declaring the reservation of products specified by the customer. Then, the assembly worker collects reserved items, manually, from stock.

The assembly team leader checks quantity and quality of each product. Then the delivery agent carries the confirmed order, so that the customer gets his ordered products.

A. Presentation of the CIM Level

Fig. 1 shows the model of the business process represented by the collaboration diagram of BPMN. In this model we just specified sub-processes and their sequence by avoiding the identification of tasks and gateways to present a business process in general. However, we have presented the maximum of collaborators to define a true business process, in which there is collaboration between several business actors, e.g. instead of putting a single lane "delivery service", we

identified the lanes: "assembly worker", "assembly team leader" and "delivery agent". This fractionation also facilitates the task of transformation to the PIM level. So when moving to the model diagram use case, collaborators will be transformed to actors. However, we presented mediums sub-processes. So the customer would normally perform the activity "select products", then "start order" and later the activity "present information", but since "start order" cannot contain more than three tasks, we 'have merged with "select products" as a single sub-process called "choose products for order". Finally, we specified all manual tasks. We can make several refinements on an initial model to achieve a model that respects our rules.

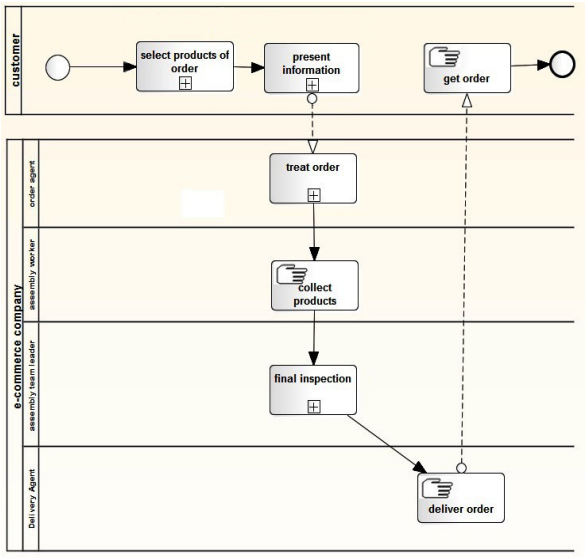

Fig. 1 Collaboration diagram model of “sales through e-commerce”

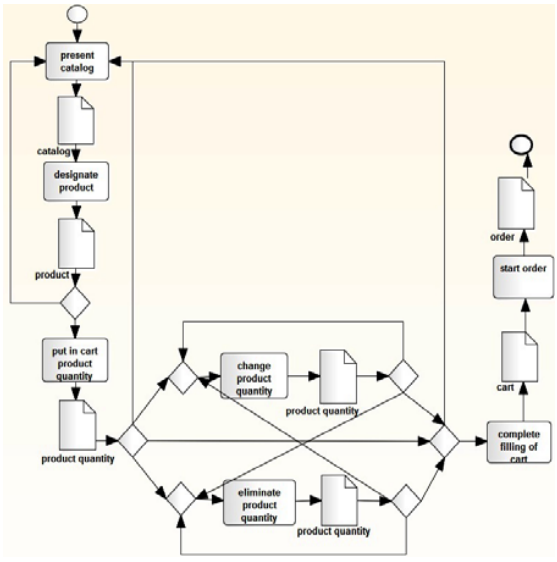

Fig. 2 Process diagram model of “select products for order”

Fig. 2 shows the second model in CIM level as a model of process diagram. Through this model we individually detail each sub-process, of the previous model, as several tasks. However, in this model the sub-processes "select product for order" is analyzed. Also, we have identified all possible paths. Then we presented a data object in the output of each action.

B. Presentation of the PIM Level

Fig. 3 presents a model of diagram use case. This model is transformed from the business models of CIM level. However, in this model the sub-process “select product for order” model of collaboration diagram of BPMN transforms to a package. Then the collaborator “customer” who performs the sub-processes becomes actor. Then the tasks that detail the sub-processes in the model of the process diagram are transformed to use cases. Exclusive gateways that lie between two tasks become relationship “extend”. For example, in this model there is an Exclusive gateway between the two actions “designate product” and “put in cart quantity product”, so the two use cases corresponding connect via an “extend” relationship. Sequence flow that lie between two actions become relationship “include”. Thus, in this model there are sequence flow between the two actions "present catalog" and "designate product," so the two use cases corresponding connect via an "include" relationship. However, it is not present in this model the sequence flow which returns backward. For example, the relationship between the task “put in cart product quantity” and “present catalog” is not specified in this model, so as not to complicate the model, and since the diagram use case focuses only on the identification of features.

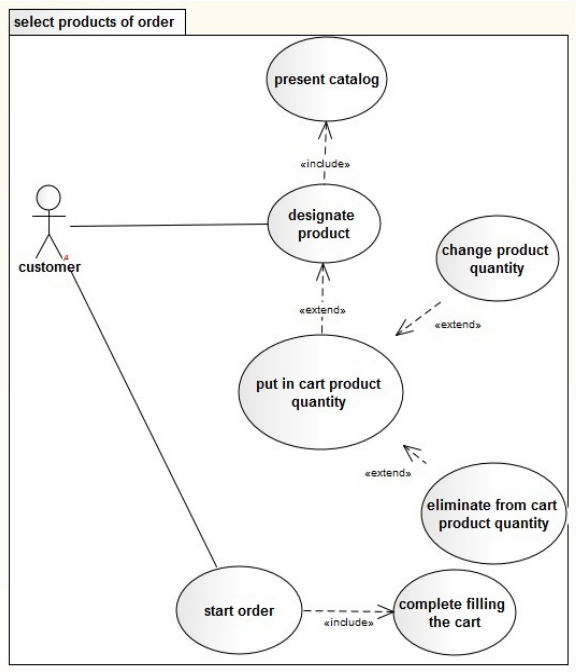

Fig. 3 Use case diagram Model of “choose products for order”

Fig. 4 shows the final objective of the PIM level is the construction of a model of class diagram. This model is

transformed from the model of the process diagram. In this model the classes are transformed from data objects. Then the several data objects that have the same name, they are transformed to one class. So the object node “quantity product” transforms to class “quantity product”.

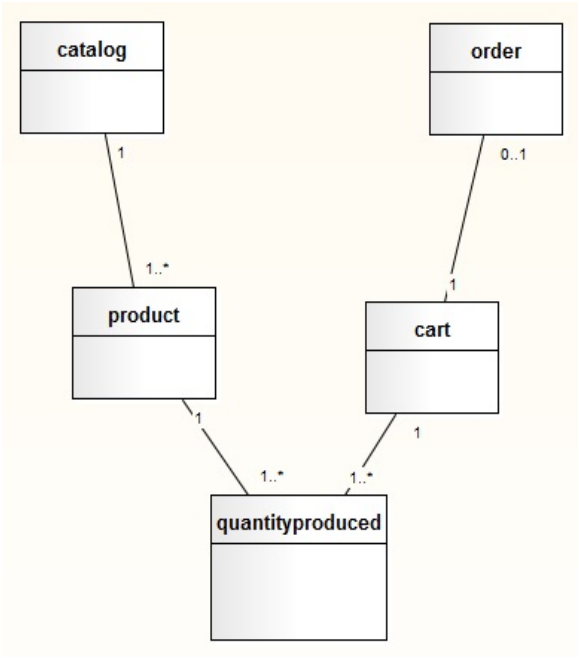

Fig. 4 Class diagram Model of “choose products for order”

V. Conclusion and Future Work

One of the major challenges in the software development process is the definition of an approach that allows moving from models that describe the operation of the business to models which present the analysis and design of software.

Based on MDA, our approach provides an efficient solution to the problem of transformation of business models represented in CIM level to analysis and design models, modeled in PIM level. This approach results a set of useful classes in the process of software development.

The ongoing work intended to improve the rules of construction of the CIM level and the rules of transformation to the PIM, In order to implement these transformations to a tool via the QVT language. In addition, in our future work, we plan to transform the models obtained in the PIM level to models of PSM level, since our ultimate goal is to provide the source code from the business models through automatic transformations.

References

[1] J. Miller, J. Mukerji, MDA Guide Version 1.0.1. Document No. omg/2003-06-01, 2003. <http://www.omg.com/mda>.
[2] OMG, UML Superstructure 2.0. OMG Adopted Specification ptc/03-08-02, 2003. <http://www.uml.org/>.
[3] OMG, Business Process Modelling Notation, Version 2.0, 2011. <http://www.omg.org/spec/BPMN/2.0/pdf>.
[4] C. Schmidt, Cover Feature Model Driven Engineering, 2006.
[5] J. Gordijn, J.M. Akkermans, Value based requirements engineering: exploring innovative e-commerce idea, Requirements Engineering Journal 8 (2) (2003) 114–134.
[6] V.D. Castro, E. Marcos, J.M. Vara, Applying CIM-to-PIM model transformations for the service-oriented development of information systems: Information and Software Technology 53 (2011) 87–105.
[7] A. Rodríguez, I. García-Rodríguez de Guzmán, E. Fernández Medina, M. Piattini, Semi-formal transformation of secure business processes into analysis class and use case models: an MDA approach, Information and Software Technology 52 (9) (2010) 945–971.
[8] S. Kherraf, E. Lefebvre, W. Suryn, Transformation From CIM to PIM Using Patterns and Archetypes : 19th Australian Conference on Software Engineering (2008) 338-346.
[9] C. Hahn, P. Dmytro, K. Fischer, A model-driven approach to close the gap between business requirements and agent-based execution, in: Proceedings of the 4th Workshop on Agent-based Technologies and applications for enterprise interoperability, Toronto, Canada, 2010, pp. 13–24.
[10] W. Zhang, H. Mei, H. Zhao, and J. Yang, "Transformation from CIM to PIM: A Feature-Oriented Component-Based Approach," presented at MoDELS 2005, Montego Bay, Jamaica, 2005.
[11] B. Grammel, S. Kastenholz, A generic traceability framework for facet-based traceability data extraction in model-driven software development, in: Proceedings of the 6th ECMFA Traceability Workshop held in conjunction ECMFA 2010, Paris, France, 2010, pp. 7–14.
[12] OMG, MOF 2.0 Query/View/Transformation (QVT), V1.0. OMG Document – formal/08-04-03, 2008.<http://www.omg.org/spec/QVT/1.0/>.
[13] J.J. Gutiérrez, C. Nebut, M.J. Escalona, M. Mejías, I.M. Ramos, Visualization of use cases through automatically generated activity diagrams, in: 11th international conference on Model Driven Engineering Languages and Systems, 2008.
[14] J. Mazón, J. Pardillo, J. Trujillo, A model-driven goal-oriented requirement engineering approach for data warehouses, in: Proceedings of the Conference on Advances in Conceptual Modeling: Foundations and Applications, ER Workshops, Auckland, New Zealand, 2007, pp. 255–264.
[15] OMG, Service Oriented Architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS). OMG document: ad/2008-08-04, 2009. <http://www.omg.org/docs/ad/08-08-04.pdf>. (Revised Submission).
[16] M. Kardoš, M. Drozdová, Analytical Method of CIM to PIM Transformation in Model Driven Architecture (MDA) : JIOS, VOL. 34, NO. 1 (2010), PP. 89-99.