Disciplined Approach for Transformation CIM to PIM in MDA

Yassine Rhazali[1], Youssef Hadi[2] and Abdelaziz Mouloudi[1]
[1]ACIDROS Laboratory, Faculty of Science, Kenitra, Morocco
[2]LaRIT Laboratory, Faculty of Science, Kenitra, Morocco

Keywords: Model Transformation, MDA, CIM, PIM, Business Processes.

Abstract: This paper suggests a disciplined approach to mastered transformation of CIM level to PIM level in accordance with the MDA approach. Our suggestion is founded on creating good CIM level through well-selected rules, allowing us to achieve rich models that contain relevant information to facilitate the task of the transformation to the PIM level. We specify, thereafter, an appropriate PIM level through different UML points of view (functional, dynamic and static) using a diagram for each one. Next, we present a set of well-defined rules to shift CIM to PIM so as to ensure an automatic transformation, the maximum possible. Our method follows the MDA approach by considering the business dimension in the CIM level, through the use standards modelling business of OMG (BPMN and Activity Diagram), and by using the UML in PIM advocated by MDA in this level.

1 INTRODUCTION

Transformations between the different levels of MDA(Model Driven Architecture) (Siegel, 2014) 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 contained in the CIM models into PIM models which would 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. The purpose is 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 automate the transformation from CIM level at PIM level, by effectively using current business modeling standards, so as to achieve focused CIM models to simplify the transformation to PIM, then, we define a set of rules to automate the conversion to a PIM level.

Our approach uses the BPMN (OMG, 2014) collaboration diagram and the UML (OMG, 2013) activity diagram which represent standards of business model to define the CIM level. Then the rich business models of well-concentrated information help us to achieve models of PIM level. The first model of the PIM level is the use case diagram that defines the functionality of the information system; then, the system states are presented through the state diagram. Next, the class diagram model allows modeling the system classes and their relationships independently of a programming language in particular. Finally, all classes are structured in packages that are transformed from the CIM level.

The rest of this paper is organized as follows. Section II analyzes the related works 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.

Rhazali Y., Hadi Y. and Mouloudi A..
Disciplined Approach for Transformation CIM to PIM in MDA.
DOI: 10.5220/0005245903120320
In *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development* (MODELSWARD-2015), pages 312-320
ISBN: 978-989-758-083-3

2 RELATED WORK

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

The oriented service transformation from CIM to PIM was proposed by (Castro ,Marcos and Vara, 2011). The authors present the CIM level using BPMN for modeling business process and the value model (Gordijn and Akkermans, 2003) 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-automatic manner with well-defined rules. But the authors' study is only limited to the use case diagram and the activity diagram in the PIM level and does 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 causes 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, García-Rodríguez de Guzmán, Fernández Medina and Piattini, 2010). The authors use the BPMN notation for modeling a secure business processes of the CIM level; then, they determine the transformations in QVT in order to obtain class diagrams and use case. This method presents a reference in the transformation of CIM to PIM security oriented. However, this proposal focuses only on secure information systems.

(Hahn, Dmytro and Fischer., 2010) 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. The authors use SoaML, the new OMG standard for modeling services, but this approach does not represent the ultimate goal of PIM level, such class digrams.

(Zhang, Mei, Zhao and Yang, 2005) describe an approach in which the CIM and PIM are respectively represented by functionalities and components. Responsibilities in this approach are considered as connectors between functionalities and components to simplify the task of transforming CIM to PIM. (Grammel and Kastenholz, 2010) 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 which aims at transforming the diagram of use case to the activity diagram is proposed by (Gutiérrez, Nebut, Escalona, Mejías and Ramos, 2008). 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, the authors define in the CIM level functional requirements represented by the use case.

(Mazón, Pardillo and Trujillo, 2007) 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, Lefebvre and Suryn, 2008) 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, then a detailed activity diagram which defines the system requirements which 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.

(Kardoš and Drozdová, 2010) 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 they 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 works concerning the passage of the CIM level to the PIM level, we can classify the works into five categories. We find works that use model requirements (such as the use case diagram) in the CIM level, to facilitate the transformation to PIM (Gutiérrez, Nebut, Escalona, Mejías and Ramos, 2008) and (Kherraf, Lefebvre and Suryn, 2008). Then, other researches (Castro ,Marcos and Vara, 2011) and (Hahn, Dmytro and Fischer., 2010), 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 target transformation in a particular field (Rodríguez, García-Rodríguez de Guzmán, Fernández Medina and Piattini, 2010) and (Mazón, Pardillo and Trujillo, 2007). There are also methods such as (Kardoš and Drozdová, 2010) which represent the structural view in the PIM level and are not intended for a particular area, but the authors do not specify transformation rules. Finally, there are methods (Zhang, Mei, Zhao and Yang, 2005) and (Grammel and Kastenholz, 2010) that define exactly 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 as follows: 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.

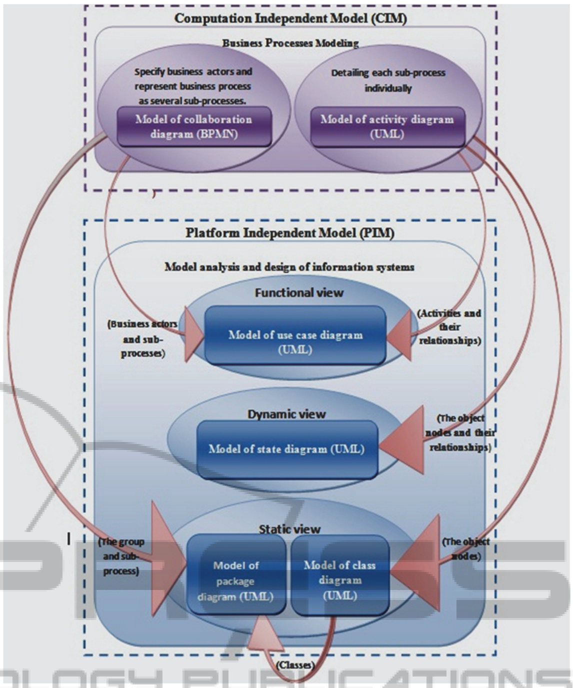

Figure 1: Schema of the proposed approach.

3 PROPOSED METHOD OF TRANSFORMATION FROM CIM TO PIM

All models of the PIM level are realized through an automatic transformation of CIM level (cf. Figure 1), 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.

3.1 Construction Rules of CIM Level

- Define means and not complex sub-processes, i.e., each process 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 of the representation of tasks, and present only manual tasks.
- The model does not describe all possible cases and ways, 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 talk about an enterprise process.
- Use the "Group" notation to group sub-processes that belong to the same category.
- Avoid in this level, the maximum possible, representation of the connections.

The rules of construction of model activity diagram:

- Detail individually each sub-process in a model as several actions (this latter constitute the fundamental unit in the activity diagram).
- Do not represent the manual tasks of model collaboration diagram.
- Present connections in this model.

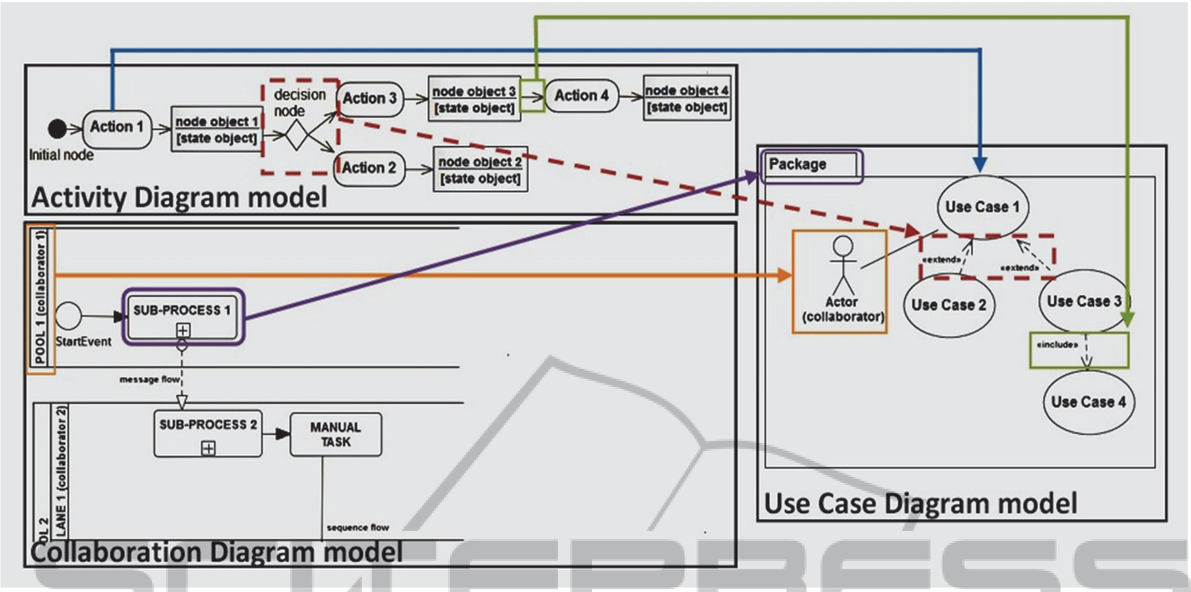

Figure 2: Schema of passage from the CIM level models to use case diagram model.

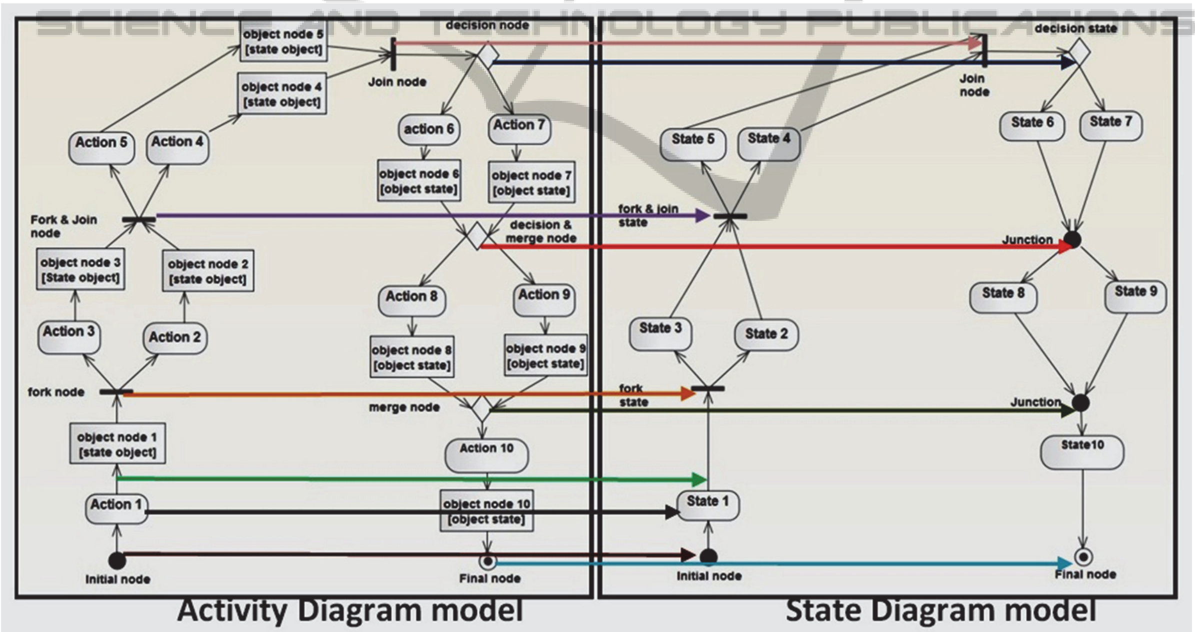

Figure 3: Schema of passage from activity diagram model to state diagram model.

- Enrich this model with the most exceptional ways.
- Any task described in the BPMN diagram model is represented here by an action.
- Add an object node containing object state at the output of each action.

3.2 Transformation Rules from CIM to PIM

The rules of passage from the CIM level models to model of use case diagram (cf. Figure 2):

- Every action of the model activity diagram that corresponds to a functionality of the system is transformed to the use case.
- The collaborator, who realizes the sub-processes of the model of collaboration

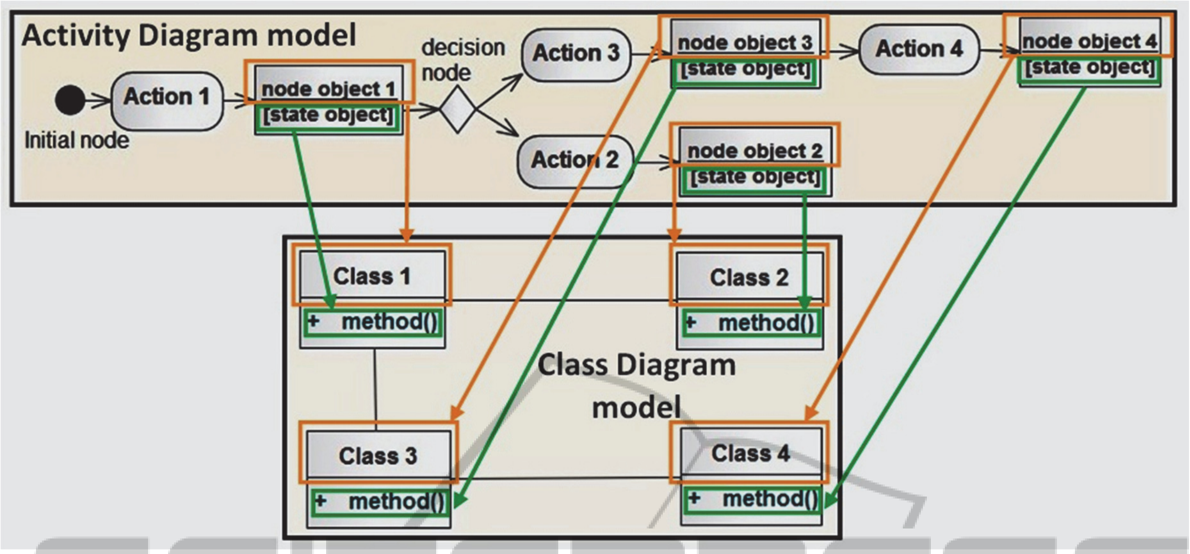

Figure 4: Schema of passage from activity diagram model to class diagram model.

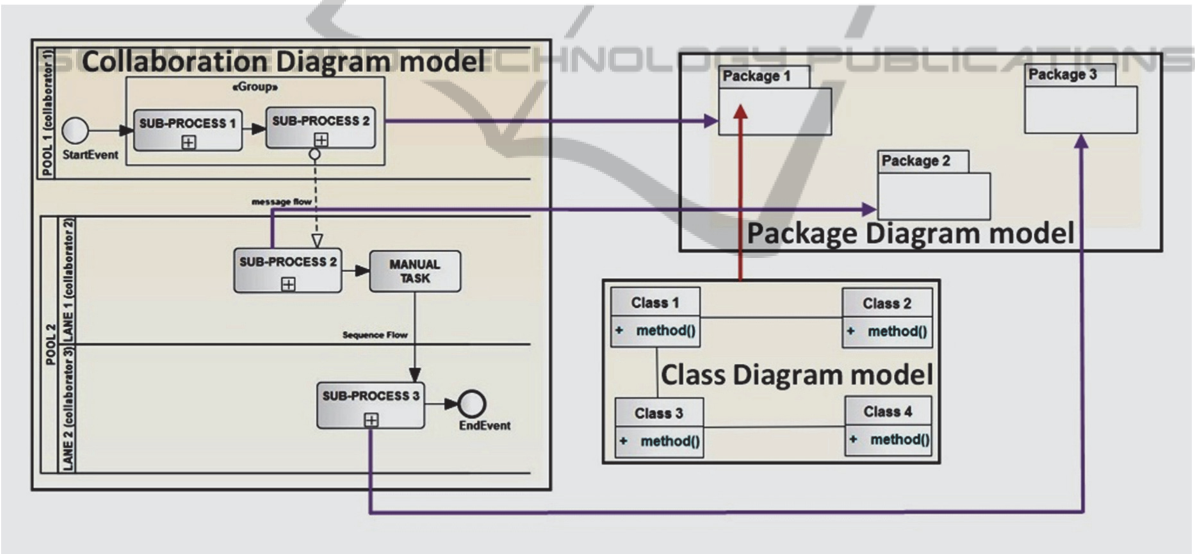

Figure 5: Schema of passage from collaboration diagram model and class diagram model to package diagram model.

diagram BPMN, becomes an actor of use cases that correspond to the actions of this sub-process.

- If there is a "decision node" between two actions, the corresponding use cases are connected by a relationship "extend".
- If there is just a control flow between two actions, the corresponding use cases are connected by a relationship "include".
- Do not transform the control flow returning back.
- Each sub-process of collaboration diagram model is transformed to a package which includes the use cases corresponding to the actions of this sub-process.

The rules of passage from model of activity diagram to model of state diagram (cf. Figure 3):

- Each node object becomes a state.
- Each decision node becomes a decision point.
- Each merge node becomes a junction point.
- Each decision and merge node becomes a junction point.
- Each initial node is transformed to an initial state.
- Each final node becomes a final state.
- Each control flow located between two actions

is transformed to a transition.

- Each fork node becomes a fork state.
- Each joint node becomes a joint state.
- Each joint and fork node becomes a joint and fork state.

The rules of passage from the model of activity diagram to the model of class diagram (cf. Figure 4):

- Transform object nodes of model activity diagram as classes.
- Each state of an object becomes a class method.

The rules of passage from the model of collaboration diagram and the model of class diagram to the model of package diagram (cf. Figure 5):

- Each group becomes a package.
- Each sub-process that does not belong to any group transforms to a package
- Each set of classes, which become the same group, will be placed in the package that matches the group.
- Classes resulting from the same sub-process, which belongs to no group, will be placed in the package that corresponds to the sub-processes.

4 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 either 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 selected, the latter starts the command. Then, he presents 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.

4.1 Presentation of the CIM Level

Figure 6 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 connections 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. For example, instead of putting a single lane "delivery service", we identified the lanes: "assembly worker", "assembly team leader" and "delivery agent".

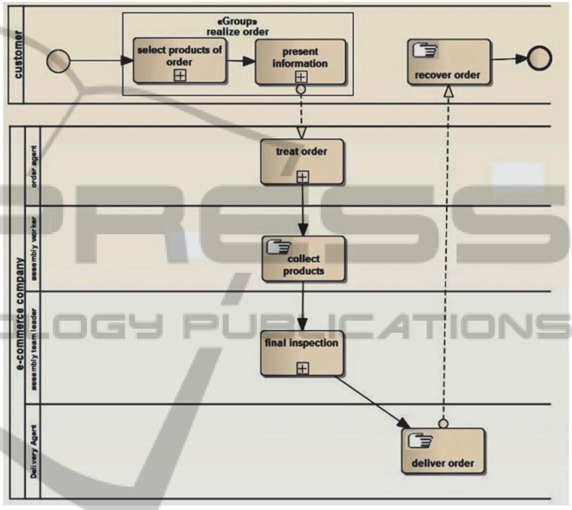

Figure 6: Collaboration diagram model of “sales through e-commerce”.

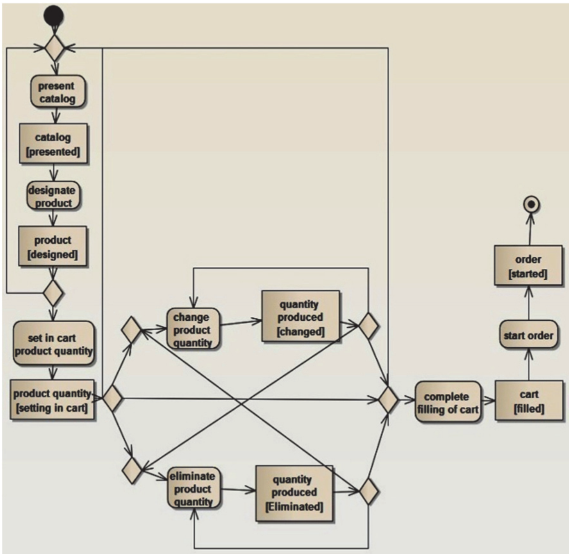

Figure 7: Activity diagram model of “select products for order”.

This fractionation also facilitates the task of transformation to the PIM level. For instance, when moving to the model diagram use case, collaborators will be transformed to actors. However, we presented medium 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 "select products" as a single sub-process called "choose products for order". Finally, we specified all manual tasks.an make several refinements on an initial model to achieve a model that respects our rules.

Figure 7 shows the second model in CIM level as a model of activity diagram. Through this model we individually detail each sub-process of the previous model as several actions. However, in this model the sub-processes "select product for order" is analyzed. Also, we have identified all possible ways towards connections. Then we presented an object node with its state in the output of each action.

4.2 Presentation of the PIM Level

Figure 8 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- is transformed to a package. Then the collaborator "customer" who performs the sub-processes becomes actor. Then the actions that detail the sub-processes in the model of the activity diagram are transformed to use cases. Decision nodes that lie between two actions become relationship "extend". For example, in this model there is a decision node between the two actions "designate product" and "put in cart quantity product "; so the two correspondent use cases are connected via an "extend" relationship. Control flows that lie between two actions become relationship "include." Thus, in this model there is flow control between the two actions "present catalog" and "designate product," so the two corresponding use cases are connected via an "include" relationship. However, it is not presented in this model the flows which return backward. For example, the relationship between the action "put in cart product quantity" and "present catalog" is not specified in this model, so as not to complicate the model, and so that the diagram use case would not focus only on the identification of functionality and not on the sequence.

Figure 9 presents the second model of the PIM level which is a model state diagram transformed from the activity diagram of the CIM level. In this model the nodes of objects are transformed to states. Then, the control flows that lie between two actions are transformed into a transition. E.g. the object node "catalog" with the state "presented" becomes "catalog presented" in the state diagram. However, the initial node is transformed to an initial state; the final node becomes a final state; the decision nodes are transformed to decision points; nodes fusion become junction points and a decision and fusion node become a junction point.

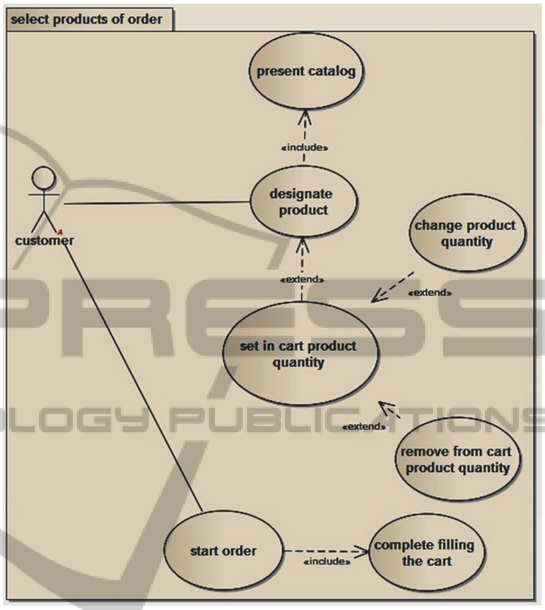

Figure 8: Use case diagram Model of “choose products for order”.

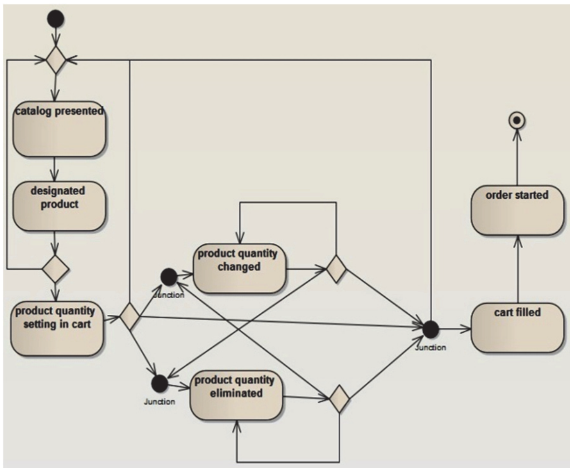

Figure 9: State diagram Model of “choose products for order”.

Figure 10 shows the final objective of the PIM level which is the construction of a model of class diagram. This model is transformed from the model of the activity diagram. In this model the classes are transformed from object nodes. Then the states of an object are transformed to functions of the class. So the object node "order" with state "started" transform to class "order" that contains the "start" method.

Figure 11 shows a model of the package diagram. So the group "realize order" transform into package. Then the sub-processes that are not in a group, such as "treat order" and "final inspection" become as packages.

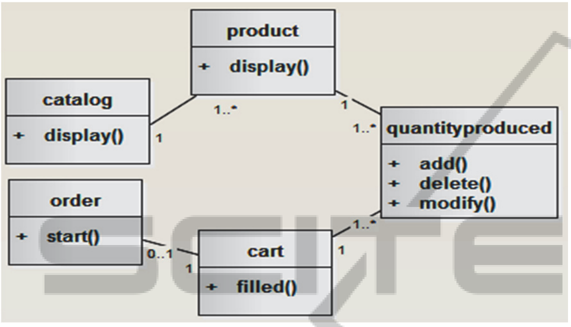

Figure 10: Class diagram Model of “choose products for order”.

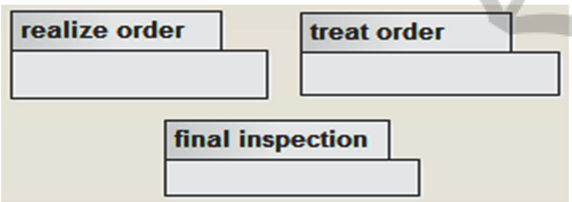

Figure 11: Package diagram model of “sales through e-commerce”

5 CONCLUSIONS 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, allowing to sharing design between people and computers. Based on MDA, our approach provides a 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 in a set of well organized and useful classes in the process of software development. The ongoing work is 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

Siegel, J., 2014. MDA Guide, revision 2.0, <http://www.omg.org/cgi-bin/doc?ormsc/14-06-01>.

OMG, 2013. Unified Modeling Language, Version 2.0.2 Beta 2, <http://www.omg.org/spec/UML/2.5/Beta2/>.

OMG, 2014. Business Process Modelling Notation, v 2.0.2, <http:// www.omg.org/spec/BPMN/2.0/pdf>.

Gordijn, J., Akkermans, J.M., 2003. Value based requirements engineering: exploring innovative e-commerce idea, Requirements Engineering Journal 8 (2) (2003) 114–134.

Castro, V.D., Marcos, E., Vara, J.M., 2011. Applying CIM-to-PIM model transformations for the service-oriented development of information systems: Information and Software Technology 53 87–105.

Rodríguez, A., García-Rodríguez de Guzmán, I., Fernández Medina, E., Piattini, M., 2010. 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.

Kherraf, S., Lefebvre E., Suryn W., Transformation From CIM to PIM Using Patterns and Archetypes. In 19th Australian Conference on Software Engineering (2008) 338-346.

Hahn, C., Dmytro, P., Fischer, K., 2010. 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.

Zhang, W., Mei, H., Zhao, H., Yang, J., 2005. "Transformation from CIM to PIM: A Feature-Oriented Component-Based Approach," presented at MoDELS 2005, Montego Bay, Jamaica, 2005.

Grammel, B., Kastenholz, S., 2010. 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.

Gutiérrez, J.J., Nebut, C., Escalona, M.J., Mejías, M., Ramos, I.M., 2008. Visualization of use cases through automatically generated activity diagrams. In 11th international conference on Model Driven Engineering Languages and Systems.

Mazón, J., Pardillo, J., Trujillo, J., 2007. A model-driven goal-oriented requirement engineering approach for data warehouses. In Proceedings of the Conference on Advances in Conceptual Modeling Workshops,

Auckland, New Zealand, 2007, pp. 255–264.
Kardoš, M., Drozdová, M., 2010. Analytical Method of CIM to PIM Transformation in Model Driven Architecture (MDA) : JIOS, VOL. 34, NO. 1 (2010), PP. 89-99.