CIM to PIM Transformation in MDA: from Service-Oriented Business Models to Web-Based Design Models

Yassine Rhazali\textsuperscript{1}, Youssef Hadi\textsuperscript{2} and Abdelaziz Mouloudi\textsuperscript{3}

MISC Laboratory, Faculty of Science Kenitra, Ibn Tofail University, Kenitra, Morocco
\textsuperscript{1}dr.yassine.rhazali@gmail.com, \textsuperscript{2}hadiyoussf@gmail.com, \textsuperscript{3}mouloudi_aziz@hotmail.com

Abstract

Models transformation is the main key of MDA. The first transformation in MDA is CIM to PIM transformation, the second is PIM to PSM transformation. Most searches deal the transformation from PIM level to PSM level, since there are multiple common points between these two levels. However, the transformation from CIM level to PIM level is rarely addressed in search subjects because they are two distinct levels. Our objective in this paper is to represent an approach that allows controlling transformation from CIM level to PIM level in accordance with the MDA approach. More precisely, we propose a methodology for transforming service-oriented business models, to web-based design models. Despite the importance of service-oriented models and web-based models, the transformation between them is not addressed in MDA researches. Our methodology is based on creating a good CIM models service-oriented, through construction rules, to facilitate transformation towards PIM models web-based. Next, our transformation rules allow a semi-automatic transformation from CIM to PIM. Our approach conforms to MDA recommendations, because it allows considering the business dimension in the CIM level, and it allows modeling this latter level by using SoaML, the OMG standard for modeling services. However, we based on UML 2 to model PIM level, because UML is advocated by MDA in PIM level. Our proposal results a set of web-based design models from service-oriented business models, through semi-automatic transformation in accordance with MDA approach.

Keywords: Model Transformation, MDA, CIM, PIM, Business Model, BPMN, SoaML, service-oriented, web-based, Computer Model

1. Introduction

The principle key of MDA (model driven architecture) [34] is the use of models in different phases of a software development cycle. Specifically, MDA recommends developing business models (CIM), analysis and design models (PIM) and code models (PSM). The major goal of MDA is developing sustainable models (CIM and PIM models), independent technical details of implementing platforms (PHP, .Net...), to allow the automatic generation of all the application code and obtain a significant gain in productivity.

The first thing to do when building a new application is to specify the company's business processes. The objective is to create a business process model, CIM (computation independent model) model, of the future application. This model should represent the application in its environment to define the services offered by the application and the entities that interact in the application. The role of business process models in an MDA approach is to be the first sustainable models. Once the modeled services, they are expected to provide a contractual basis, because validated by the
application client. With traceability links with the other models, a link can be created from business processes to the Analysis and design.

Analysis and design are the existing modeling stages from a long time. The design is the step of structuring the software into modules and sub-modules. In PIM (platform independent model) level, we consider only the abstract design, one that is done without any knowledge of implementation techniques. The role of analysis and design models, PIM models, is to be sustainable and to make the link between business models, CIM models, and code models (PSM models).

Once the analysis and design models made, the code generation work can begin. The main difference between a code model and an analysis or design model resides in the fact that the code model is related to a platform of execution. In the MDA vocabulary, these code models are called PSM (Platform Specific Model).

MDA highlights the importance of model transformations, but, does not propose any methodological transformation process. Most searchers founded on the transformation from PIM level to PSM level, since there are several links between these levels. But the transforming from CIM level to PIM level is rarely discussed in research topics because they are two dissimilar levels. For that, in this research we propose a methodology that allows mastering transformation from service-oriented CIM level to web-based PIM level.

The remainder of this paper is presented as follows: In Section 2, we represent the related works concerning transformation from the CIM level to the PIM level. In Section 3 we show our proposal by describing the construction rules of the CIM models and the transformation rules that allow moving from CIM level to PIM level. An illustration of our approach in a case study is represented at Section 4. Section 5 discusses the performance evaluation of our transformation approach. Finally, we conclude by specifying the future work in Section 6.

2. Related Work

In this section, we describe related works concerning transformation approaches from CIM level to PIM level into MDA.

Kherraf et al., [1] represent a method based on patterns and archetypes for transforming CIM models to PIM models . The authors use patterns to structure CIM level, and archetypes to move toward PIM level. This approach founded on two stages for modeling the CIM level. The first stage bases on UML 2 [35] use case diagram and activity diagram for modeling business process. The second stage bases on activity diagram to model system requirements. Then, requirement elements transformed to system components into the first model in PIM level. Finally, a set of four archetypes [17, 18] helps to transform system components to class diagram model.

Zhang et al., [2], present a feature-oriented and component-based method for transforming CIM models to PIM models. The authors found on the feature model for structuring requirements in CIM level. This latter model contains features and their relationships. Software architecture model represents PIM level. This last model contains a set of components and interactions between them. However, responsibilities considered as connectors between features and components for transforming CIM models to PIM models.

Kardoš et al., [3] represent an analytical method to move from CIM level to PIM level. The authors found on the Data Flow Diagram (DFD) [23, 19], to represent business process in CIM models. Nevertheless, PIM models founded on four UML diagrams includes use case diagram, activity diagram, domain model, and sequence diagram.

Rodríguez et al., [4] propose a transformation approach from secure business process to use cases. This approach presents business process in CIM level through secure business process model which founded in BPMN notation [36]. Nevertheless, a set of transformation rules, checklists, and refinement rules allows transforming secure business
process model to use case model that represent PIM level. In [5, 6] Rodríguez et al., founded on the same previous methodology but UML 2 activity diagram used with BPMN notation to model the secure business process, and PIM level enlarged by the class diagram model.

A transformation methodology from CIM to PIM was proposed by De Castro et al. [7]. The authors use BPMN for modeling the business process, and value model [20] for identifying services in CIM level. However, PIM models are defined by two extensions of UML activity diagram and two extensions of UML use case diagram.

A transformation methodology from business requirements towards agent-based execution is shown by Hahn et al. [8]. The authors apply BPMN notation to define CIM models. However, SoaML [21] model used to model agents in PIM level. This approach based on some for moving from CIM level to PIM level.

A methodology for transforming model-driven goal-oriented requirement to data warehouse is shown by Mazón et al. [9]. The authors represent CIM level with UML profile using the i* modeling framework [22]. However, some rules used to move from CIM level to PIM level which is represented by data warehouse design.

An approach where Activity diagram model generated automatically from use case model is represented by Gutiérrez et al. [10]. The authors propose an approach for generate a transformation from system requirements model represented by use case diagram, into CIM level, to activity diagram model which interprets PIM level.

A transformation approach from business process model to Information system model is shown by Mokrys [11]. The author apply BPMN notation in business process model for representing CIM level. Nevertheless, PIM models are presented through UML 2 state diagram and class diagram.

An approach allows modeling CIM and transforming it to PIM is shown by Bousetta et al. [12]. The authors represent CIM with low level business model, and use case model. However, PIM models are defined by sequence diagram, and domain class diagram. The class model is transformed from low business process model through business rules and objects, resource.

Fatolahi et al., [13] show a methodology of semi-automatic transformation from use cases to web-based applications in accordance with MDA. The authors interpret CIM level with requirement model through use cases and the default domain objects. Nevertheless, PIM level is defined by state machine model, user interface model, and refined domain model.

A method for establishing CIM models and transforming them to PIM models according MDA is presented by Wu et al., [14]. The author based on use case diagram, robustness diagram, and activity diagram for modeling user requirements in CIM level. Nevertheless, PIM models are represented by UML 2 class diagram and sequence diagram.

In our previous research [15], we have proposed a method to transform business process models to information system models. The method is founded on BPMN notation to model business process into CIM level, however, the PIM models is presented by class diagram and use case diagram.

In another work [24], we have presented a transformation methodology from CIM level to PIM level. This methodology present CIM level through business process models by using UML activity diagram and BPMN notation. The PIM models are presented by state diagram, class and package diagram. This approach is based on a set of transformation rules that allows moving from CIM to PIM.

In [16] we have presented an approach for transforming CIM to PIM. This methodology is founded on BPMN notation and UML activity diagram for modeling business process. The PIM models are established by state diagram, use case diagram, and class diagram. This approach is based on an improved method for shifting from CIM level to PIM level.
We also presented another approach for transforming CIM level to PIM level [25]. This approach is based UML 2 activity diagram to model business process in CIM level. The PIM models are presented by class diagram, state diagram, and package diagram. This approach is founded on transformation rules for moving from CIM to PIM.

In our work in [26], we have presented a transformation approach from CIM level to PIM level. The method is based on BPMN notation to model business process into CIM level. The PIM models are presented by use case, state diagram, class and package diagram. This approach is based on a set of improved transformation rules that allows shifting from CIM models to PIM models.

3. Proposed Method of Transformation from CIM to PIM

Computation Independent Model (CIM) means that this model does not contains any information about the software system. According to OMG [27], CIM level is presented by business process models which are supposed to be representative of the real world. According OMG in [21] “SoaML services architecture shows how multiple participants work together, providing and using services to enable business goals or processes”.

In our approach, we modeled business process in the CIM level through SoaML. CIM level is presented by general business model, through SoaML service architecture, and detailed business model, through SoaML service choreography. According OMG in [21] SoaML service choreography can be presented by activity diagram or sequence diagram, in our approach we present it through activity diagram. The most interesting of our methodology is the consideration, at the beginning, that we are in the process of creating business models service-oriented that will be automatically transformed to information system models. We carefully specified a set of rules to define business models, CIM models, which help us to achieve an easy transformation. Our approach keeps the business knowledge during the transformation to the PIM. This allows us to product a quality information system.

We divided PIM models according to four views including web view and the three classical modeling views [28]-[30]: functional, static, dynamic view. Indeed, many researches declare that UML is recommended by MDA in the PIM level as in [31, 32]. According [21] the SoaML is an extension to UML 2 to support service concepts. Always according [21] SoaML enables business oriented and systems oriented services architectures to mutually and collaboratively support the enterprise mission. However, the intersection of our UML 2 models and the four views is presented as follows: The model of the use case diagram interprets functional view; the model of the state diagram represents the dynamic view. The models of class and package diagram show the static view. The web view is represented by SoaML and web modeling diagram (Figure. 1).
Figure 1. CIM to PIM Transformation Approach

The transformation from service-oriented CIM level to web-based PIM level is based on well concentrated rules. In the following subsections, we represent our transformation rules.

3.1. Construction Rules of CIM Level

The rules of construction the general business model through SoaML service architecture (Figure. 2):

- Define medium services; in fact, each service must contain between 4 and 10 actions.
- If a service contains less than 4 actions, or represents an additional operation to another service, we can merge various services into one, provided that, the service does not exceed 10 actions.
- Show manual service with gray color.
- The model represents the general business processes.
- Based only on services, participants and their relations.
- Show the maximum of the business participants which collaborate in the realization of enterprise business processes.

The rules of construction the detailed business model through SoaML service choreography (Figure. 2):

- Detailing individually each service as various actions (the action constitutes the fundamental unit in the SoaML service choreography).
- Represent gateways in this model.
- Show the most exceptional paths.
- Add a data object containing object state at the output of each action.
3.2. Transformation Rules from CIM to PIM

Transformation rules from general and detailed business model to use case model

The (Figure. 3):

- Every action, of detailed business model, transformed to use case.
- The participant, of general business model, becomes an actor.
- A “decision node”, of detailed business model, becomes a relationship “extend”.
- Do not transform control flow that returns back, of detailed business model.
- A control flow, of detailed business model, becomes a relationship “include”.
- Each service, of general business model, transformed to a use case package.

Transformation rules from detailed business model (SoaML service choreography) to state diagram model (Figure. 4):

- A state of the object node becomes a state.
- A decision node becomes a decision point.
- A merge node becomes a junction point.
- A decision and merge node becomes a junction point.
- An initial node becomes an initial state.
- A final node becomes a final state.
- A control flow becomes a transition.
- A fork node becomes a fork state.
- A joint node becomes a joint state.
- A joint and fork node becomes a joint and fork state.

Figure 2. Generic Schema of CIM: I General Business Model, II Detailed Business Model
Figure 3. Schema of Passage from SoaML Business Models to Use Case Diagram Model

Figure 4. Schema of Passage from Detailed Business Model to State Diagram Model

Transformation rules from detailed business model (SoaML service choreography) to class diagram model (Figure. 5):

- An object node becomes class.
- A state of an object node becomes a class method.
Transformation rules from general and detailed business model to package model (Figure. 6):

- A package, of general business model, becomes a package.
- A service, of general business model, which does not belong to any package, becomes a package.
- A set of classes, which becomes from the same package, will be placed in the package corresponding to the package of general business model.
- Classes coming from the same service, that does not located in any package of general business model, will be placed in the package matching to the service.
Transformation rules from design models, state, class, and package diagram model, to SoaML software components model (Figure 7):

- A class transforms to an entity.
- A state transforms to request point.
- A state transforms to service point into entity.
- A package becomes service contract
- Request points must be presented into web site
- Service point must be presented into entity

Figure 7. Schema of Passage from Activity, Class, and Package Diagram Model to SoaML Software Components Model

Transformation rules from design models, use case, state, and class diagram model, to web modeling diagram model (Figure 8):

- A class transforms to a web page name.
- If the class has several methods, each method transforms to a page web.
- Each web page have pre-condition and post-condition.
- Each state becomes a post-condition in the web page.
- If the web page has in the direct previous input a link, the pre-condition of the web page will be “name page + chosen”.
- If the web page has in the direct previous input a submit, the precondition of the web page will be “confirmed + name of realized functionality in the previous web page (use case that corresponds the previous web page)”.
- The sequencing of the web pages must match the sequencing of the states in state diagram model (each transition becomes link, submit, or redirect).
- The functionality, authentication, presented in use case diagram model transforms to an authentication web page to ensure the security of the information system.
- The authentication page contains two text areas elements: the login and password. This page must have as first output a “submit” liaison to the next page if the login and password are correct, however, it must have as second output a reflexive liaison “redirect” if login and password are incorrect.
- The main page should be added as a first web page or a second web page if the authentication precedes all functionality.
- Each web page that contains a “text area element” must have in output a” submit” liaison.
• When there are multiple outputs in a page, which contains text area elements, this latter page will have a reflexive submit, while the other outputs will be interpreted as links.
• The link name is the name of the destination web page.

Figure 8. Schema of Passage from Use Case, Activity, and Class Diagram Model to Web Modelling Diagram Model

4. Case Study

In this section, the authors present a case study for sales through e-commerce to illustrate this 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 9 shows general business model represented by SoaML service architecture. In this model we specified services, participants and the roles which play each participant in the service. We have presented the maximum of participants to define a true business process, in which there is collaboration between several business actors. For example, instead of putting a single participant "delivery service", we identified the participants: "assembly worker", "assembly team leader" and "delivery agent".
Figure 9. General Business Model of SoaML Service Architecture of “sale products”

Figure 10 shows the second model in CIM level as detailed business model of SoaML service choreography. Through this model we individually detail each service contract of the previous model as several actions. However, in this model the sub of service contract “select order product” is detailed. Also, we have identified all possible ways towards connections. Then the authors presented an object node with its state in the output of each action.

4.2. Presentation of the PIM level

Figure 11 I, shows a use case diagram model. This latter model is transformed from CIM models. Nevertheless, the service contract “select product for order” becomes a package inside use case diagram model. Next, the participant “customer” that realizes the service contract becomes actor. However, the actions which detail the service contract become use cases. Indeed, Decision node which connects between two actions becomes relationship "extend" between two use cases. For example, in the detailed business model SoaML service choreography, there is a decision node that lies between the two actions "designate product" and "put in cart quantity product ", so the two correspondent use cases are connected by the relationship "extend". Control flow which connects between two actions becomes relationship “include” between two use cases. Thus, in the detailed business model, there is control flow between the two actions “present catalog” and "designate product", indeed, the two corresponding use cases are connected by a relationship "include". Nevertheless, the control flow returning backward did not transform as a relationship “include”. For example, the relationship between the actions "put in cart product quantity" and "present catalog" did not interpret in use case diagram model, so as not to complicate the model, and because the use case diagram focuses only on the identification of functionalities and not on the sequence of the functionalities.
Figure 10. Detailed Business Model of SoaML Service Choreography of Service Contract “select order product”

Figure 11. I: Use Case Diagram Model of “select order product”, II: State Diagram Model of “select order product”

Figure 11 II, presents the state model diagram transformed from the detailed business model of CIM. In this model the nodes of objects are transformed to states. Then, the control flow that lie between two actions are transformed to 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 12. Class Diagram Model of “select order product”, II: Package Diagram Model of “sale product”

Figure 12 I, shows the final objective of the PIM level which is the construction of a class diagram model. This model is transformed from detailed business model. 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 12 II, shows a model of the package diagram. So the package "realize order" transform to package in package diagram model. Then the services that are not belong to any group, such as "treat order" and "final inspection" become packages.

Figure 13. Web Modeling Digram Model of “select order products”

Fig. 13 represents model of web modeling diagram. This last model transformed from design models: use case diagram model, state diagram model, class diagram model.

Fig. 14 represents SoaML software components. This last model transformed from: state diagram model, class, and package diagram model.
5. Analysis and Discussion

In the related works about transformation from CIM level to PIM level, we can divide these works into four types. We note approaches [1, 6] that use model of system requirements (as use case diagram) early in the CIM level, to facilitate transformation to PIM level. However, other works [7, 8] even if they base on business processes in CIM level, do not establish the structural view (usually through the class diagram) in PIM level. Then, there are researches like in [3] that establish the structural view in the PIM level, but the authors do not specify transformation rules. Finally, there are approaches [2, 33] which describe precisely the transformation rules, but do not have the models used in the CIM and PIM.

Our actual method presents a first approach, according related works, of transformation from service-oriented CIM to web-based PIM. In this method we benefited from our experience in old methods [15, 16], [24]-[26], and [37]-[40] These approaches give several methodologies for transforming CIM to PIM but they are based on BPMN to create CIM level for this we are limited when we transform CIM level into the PIM level.

Concerning related work we find that the closest approach to our method is [6]. This approach allows establishing services into CIM level by basing on value model that is nor a business modeling standard, and nor service modeling standard. Nevertheless, there is no exploitation of services in the PIM level, for example implementation of services in a web model. Then the PIM level is presented just by the models of the use case diagram and the activity diagram, which makes the transformation to PSM very difficult. Indeed, in the PIM level, there are not classes on which we base to move toward code models in PSM. However, this method does not provide clear rules to transform the CIM level to the PIM level.

Our approach, according to the related work, is the unique method based on construction rules for structuring CIM in order to facilitate the transformation toward the PIM. However, the builder of CIM level must produce models intended to be transformed
to PIM, by using optionally several refinements on the base models and by respecting our construction rules of the CIM.

Approaches of related work do not provide clear and structured transformation rules. In most approaches, we do not find any description of the rules; the reader must deduce the rules from the case study. In the rest approaches there are subsections which contain just rules hints. Our approach describes clear transformation rules with graphic presentation.

Our objective in this methodology is not just the transition from the CIM level to the PIM level, but our goal is to attain a rich PIM level that can be transformed thereafter into the PSM level.

6. Conclusions

One of the principal challenges in software development process is the establishment of an approach that allows transforming semi-automatically the business process models to software design models.

Although the massive use of service-oriented models and web-based models, the transformation between them models is not yet considered in MDA researches. For that, in this paper we provided a methodology based on creating a good CIM models service-oriented, and transforming them through transformation rules, to PIM models web-based.

According to the related work our method is the alone approach founded on construction rules for structuring CIM level in order to facilitate the transformation to the PIM level. Next, the related work approaches do not provide clear and organized transformation rules; however, our method establishes clear transformation rules with graphic presentation. Then, we provide multiple models in PIM level especially class diagram model because our objective in this methodology is not just transforming the CIM level to the PIM level, but our objective is to attain a rich PIM level that can be transformed thereafter to the PSM level.

However, in the ongoing work we define a transformation from obtained PIM models to PSM models. Nevertheless, in our future work we plan to construct a transformation tool founded on MDA principles, indeed, our goal is to obtain the code automatically from the business models.

References


Authors

Yassine Rhazali, he received his Master Degree in Software Quality from the Ibn Tofail University, Faculty of Sciences Kenitra, Morocco, in 2012. He later prepares his PhD degree in the Ibn Tofail University, Faculty of Sciences Kenitra, Morocco, since 2012. Currently, he is teaching Software Engineering at the same University and at the Moroccan office of professional training and employment promotion (OFPPT). He is a member of MISC Laboratory. His research interest focuses on conceptual modeling, software process, and modernization of legacy systems. He has valuable contributions and publications in those topics of research. You may contact him at dr.yassine.rhazali@gmail.fr.

Youssef Hadi, he received his PhD degree from Mohammed V University, Faculty of sciences Rabat, Morocco in 2008. Currently, he is a research director at Ibn Tofail University, Faculty of Sciences Kenitra, Morocco and responsible of many Masters Formations sections at the same university. He is a vice-director of the MISC Laboratory and a leadership of research’s teams on Software engineering and Multimedia Database indexing. He has valuable contributions and publications in those topics of research. You may contact him at hadiyoussf@gmail.com.