Model transformation from CIM to PIM in MDA: from business models defined in DFD to design models defined in UML

Transformation de modèles CIM à PIM selon MDA : des modèles métiers définis avec DFD aux modèles de conception UML

Yassine Rhazali
MISC Laboratory, Faculty of Science Kenitra, Ibn Tofail University, Kenitra, Morocco
dr.yassine.rhazali@gmail.com

Youssef Hadi
MISC Laboratory, Faculty of Science Kenitra, Ibn Tofail University, Kenitra, Morocco
hadiyoussef@gmail.com

Abdelaziz Mouloudi
MISC Laboratory, Faculty of Science Kenitra, Ibn Tofail University, Kenitra, Morocco
mouloudi_aziz@hotmail.com

Résumé

Cette recherche représente une méthodologie qui contrôle la transformation de modèles du niveau CIM au niveau du PIM en respectant l'approche MDA. Notre méthodologie est fondée sur l'établissement d'un bon niveau CIM, grâce à des règles bien choisies, afin de faciliter la transformation au niveau PIM. Cependant, nous créons un niveau de PIM riche grâce à un modèle du diagramme de cas d'utilisation, modèle de diagramme d'états, modèle de diagramme de classes et modèle de diagramme de package. Ensuite, nous établissons des règles de transformation pour assurer une transformation semi-automatique depuis le niveau CIM vers le niveau du PIM. Notre approche est conforme à l'approche MDA en prenant en considération la dimension métier au niveau CIM, puisque, nous présentons ce niveau par DFD et par le diagramme d'activité de l'UML. Cependant, nous modélisons le niveau PIM par les diagrammes d'UML, parce que UML est recommandé par MDA à ce niveau.

Abstract

This research represents a methodology that controls model transformation from CIM level to PIM level in accordance with MDA approach. Our approach is founded on establishing a good CIM level, through well-selected rules, to facilitate transformation to PIM level. However, we create a rich PIM level through use case diagram model, state diagram model, class diagram model and package diagram models. Then, we establish transformation rules to ensure a semi-automatic transformation from CIM level to PIM level. Our Approach conforms to MDA approach by taking into consideration the business dimension in CIM level, since we present this level through DFD and UML activity diagram. However, we model PIM by UML diagrams, because UML is recommended by MDA on this level.

Mots-clés

Transformation des modèles, MDA, CIM, PIM, PSM, processus métier.

Keywords

Model transformation, MDA, CIM, PIM, PSM, business process.

1. Introduction

Model Driven Engineering (MDE) (Schmidt, 2006) 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 Model Driven Architecture (MDA) supported by OMG is presented as the most interesting and the most common variant (OMG-MDA, 2014). MDA has the same principle as MDE, but it provides its own characteristics, defined on three levels of abstraction, defines some requirements to be respected, and also requires the use of some standards.
The first level of MDA is the Computation Independent Model (CIM) presented as model used by business managers and business analysts in order to describe the business process. The second level is the Platform Independent Model (PIM), which allows defining the models used by analysts and software designers to achieve an independent analysis and the conception of the developed software. The third level is the Platform Specific Model (PSM) which is considered model of code used by software developers. These models are believed to contain all the information needed to operate an execution platform used by software developers. The code is not a model of MDA, but it is the final result of the MDA process.
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 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 be limited to documents of communication between business experts and 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 current standards of business modeling effectively so as to achieve focused CIM models to simplify the transformation into PIM. Then, we define a set of rules to automate the transformation into a PIM level.
Our approach uses the DFD and the UML (OMG-UML, 2011) activity diagrams 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. We divide the models of PIM level according to the three classical modeling views (Roques, 2004) including functional, static, and dynamic views. According to (Blanc, 2005), UML is recommended by MDA on the PIM level. Indeed, the first model of the PIM level is the use case diagram that defines the functionality of the information system, then the dynamic view is presented through the state diagram. Next, the class diagram model allows modeling static view through representing 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.
Our previous works were based on BPMN and/or Activity diagram to model business process. However, the data flow diagram (DFD) is a simple modeling standard, because it is composed of a limited number of notations, and it is used by many researchers in their works to model the business process on CIM level like in (Kardoš and Drozdová, 2010). Indeed the searchers, who do not master design with complex language like BPMN and activity diagram, can design the CIM level, in our approach, through a simple language as DFD. For that, in this approach we benefit from the simplicity of the DFD to model the business process.
The rest of this paper is organized as follows. Section 2 presents our approach and describes the rules for constructing models of CIM level and the rules for transformation from the CIM level into the PIM level. In Section 3, we illustrate our proposal in a case study showing the construction of the CIM level and the transformation into the PIM level. Section 4 analyzes the related works of the CIM transformation into PIM. Then, we represent evaluation criteria and meta-models. Next, we analyze and discuss the transformation from CIM into PIM, by studying related works and our method. Finally, in Section 7, we conclude by determining the outcome of our work and describing future works.

2. Proposed method of transformation from CIM into PIM

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

In this approach, we do not use DFD and UML 2 activity diagram just to present an approach that uses both standard modeling business processes, but we try to take advantage of each one to achieve a rich and concentrated level, which simplifies our transformation into the PIM level.
MDA recommends the use of UML on the PIM level. The use case diagram model presents the information system functionalities while class diagram shows the structure of the information system. Then, we organize all classes in packages themselves transformed from the CIM level.
Consequently, the input models in our transformation approach are DFD and activity diagram and the outputs models are use case diagram, state diagram, class diagram and package diagram.

Figure 1. Input models and output models

All PIM models in our approach are realized through an automatic transformation of CIM level, via well-defined and concentrated transformation rules; indeed, we have defined several rules in the beginning and then we filtered these rules by eliminating less powerful rules, by merging the identical rules, and by treating exceptional cases.
In our approach, CIM level is described by the business process while PIM level is represented by the three perspectives: functional view, dynamic view and static view. Then, based on the source meta-model and target meta-model, we define a set of transformation rules to move from CIM level to PIM level.
Below, we present the rules of construction of CIM level and the rules of transformation into the PIM level.

2.1 Construction rules of CIM Level

The rules for constructing the model of DFD (figure 2):

- Define means and not complex processes, i.e. each process must not contain other processes. In fact, each process must be comprised of about 4 to 12 tasks.
- Merge two processes into one if a sub-process consists of less than 4 tasks, or represents a complementary operation to another process.
- Coloring manual processes with another color; for example, we used gray.
- Verify that the model describes the most common business processes.
- Identify most actors who interact and who collaborate in the achievement of business processes since we talk about an enterprise process.

In this model, we show the processes and their relationship for modeling a general business process. This model identifies all business actors in order to show a real business process. The representation of multiple external triggers facilitates the transformation from CIM into PIM. Indeed, when moving from DFD model to the use case diagram model, the external trigger became actors. Nevertheless, we can specify average processes; for example in hotel accommodation, the customer must present the processes «choose room», «start reservation» and «present information», but the process «start reservation» holds less than four tasks; for this, we merge «choose room» and «start reservation» into a single process called «choose rooms for reservation».
The rules of construction of activity diagram model (cf. figure 2):

- Detail individually each process in a model as several actions (this latter constitute the fundamental unit in the activity diagram).
- In activity diagram model do not represent the manual tasks of DFD model.
- Present connections in this model.
- Enrich this model with the most exceptional ways through the gateways (decision node, merge node…). For example, in the case of the payment of an order, after we enter the password of our credit card, in the nominal case, the amount of order will be subtracted from our amount. But it should be noted that there is a case where the password is incorrect and another case in which the amount in account is insufficient to complete the transaction.
- Add an object node containing object state at the output of each action.

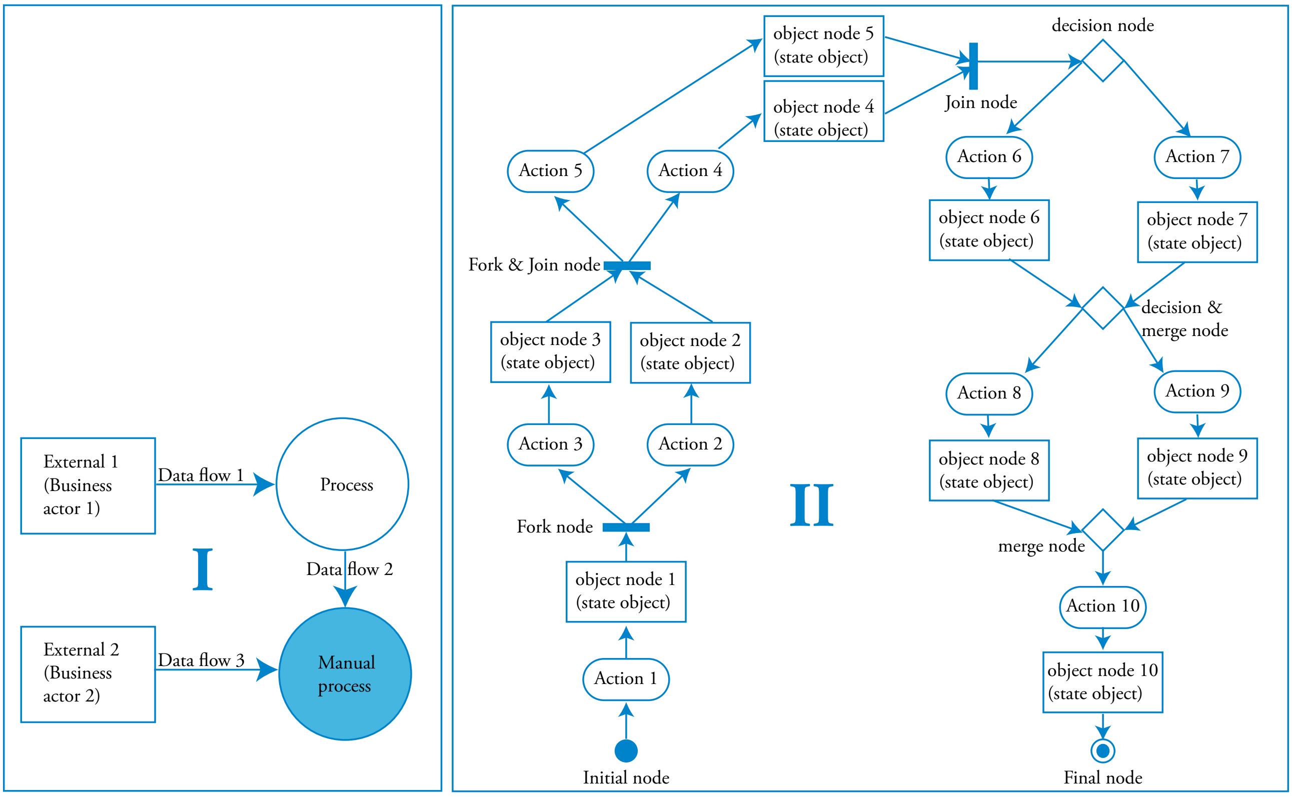

Figure 2. I: Generic model proposed of Data Flow Diagram II: Generic model proposed of activity diagram.

In this model, we represent the different actions contained in each process. For example in hotel accommodation, the process «choose room for reservation» is described by multiple actions. In the output of every task we show an object node with its state.

2.2 Transformation rules from CIM into PIM

The rules of passage from the CIM level models to model of use case diagram (Figure 3):

- Every action of the model activity diagram that corresponds to a functionality of the system is transformed into the use case.
- The collaborator, who realizes the process of the model of DFD, becomes an actor of use cases that correspond to the actions of this process.
- If there is a «decision node» between two actions, the corresponding use cases are connected by a relationship «extend». For example, Decision Point is between Action 1 and Action 2, and between Action 1 and Action 3, so the relationship «extends» will be presented between UC1 and UC2 then between UC1 and UC3.
- 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 process of DFD model is transformed into a package, which includes the use cases corresponding to the actions of this process.

For example in hotel accommodation, the process «choose room for reservation» is transformed into a package of use case diagram. Then the external «customer» transformed to actor, and the actions that detail the process «choose room for reservation» become use cases. Next, the gateway xor, which connects two actions, is transformed into a relation «extend», and the sequence flow, which connects two tasks, transforms into a relation «include». Nevertheless, we do not show the sequence flows that recur back.

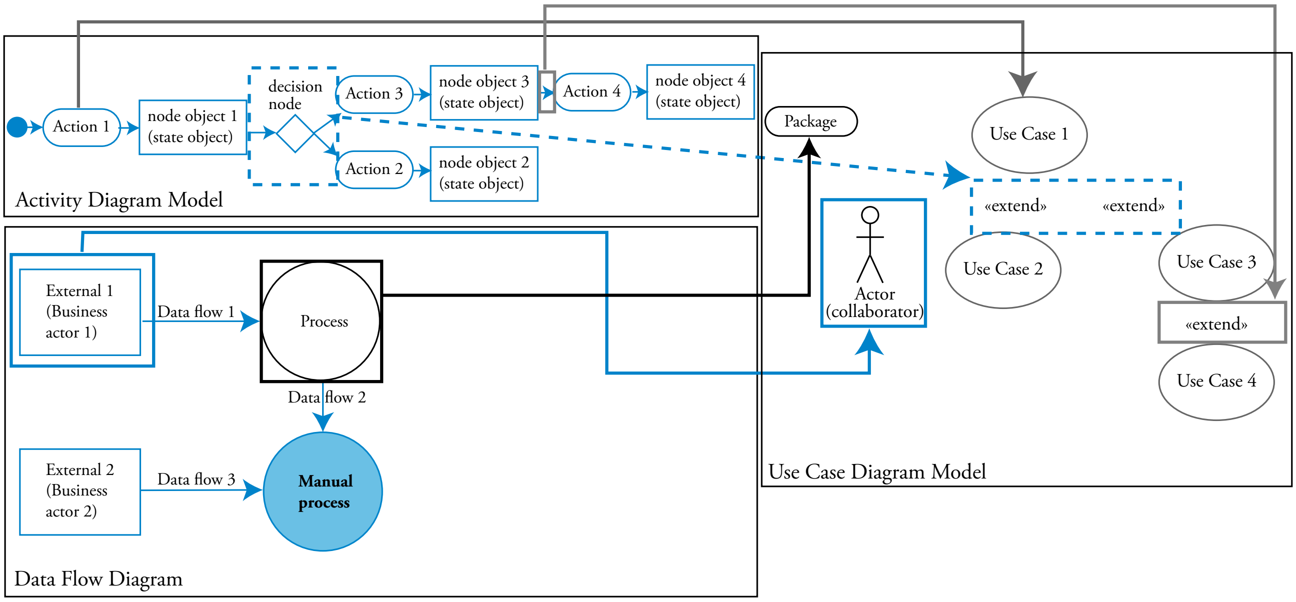

Figure 3. Schema of passage from DFD model and activity diagram model to use case diagram model

The rules of passage from model of the activity diagram to model of state diagram (Figure 4):

- R1: An object node transformed into a state.
- R2: A decision node transformed into a decision point.
- R3: A merge node transformed into a junction point.
- R4: A decision and merge node transformed into a junction point.
- R5: An initial node transformed into an initial state.
- R6: A final node transformed into a final state.
- R7: A control flow between two actions transformed into a transition.
- R8: A fork node transformed into a fork state.
- R9: A joint node transformed into a joint state.
- R10: A joint and fork node transformed into a joint and fork state.

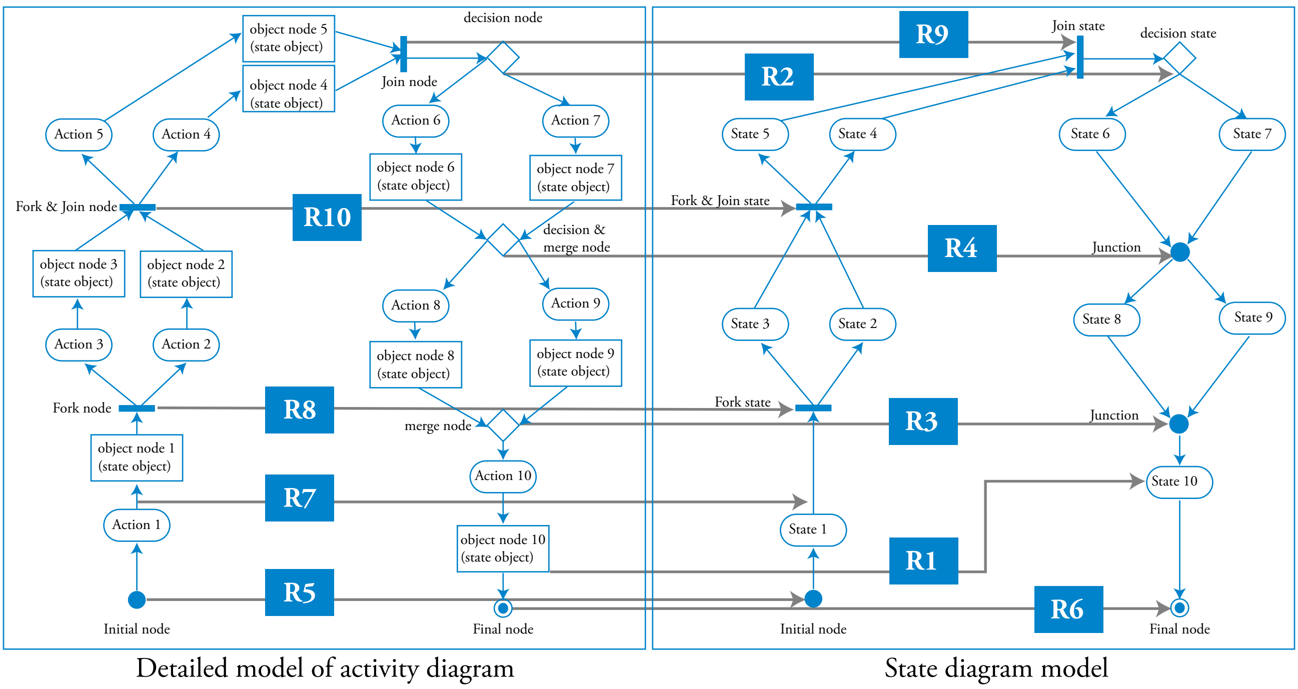

Figure 4. Schema of passage from activity diagram model to state diagram model

The state diagram model is transformed from model of activity diagram. First, object node is transformed into state and the control flow that lies between two tasks is transformed into a transition. For example, in hotel accommodation, the object node «Catalogue» with the state «displayed» becomes «Catalog displayed» in the state diagram model. However, the initial node is transformed into an initial state; the final node becomes a final state; the exclusive fork is transformed into a decision point; exclusive join become junction point; finally an exclusive fork & join node becomes a junction point.
The rules of passage from the model of activity diagram to the model of class diagram (Figure 5):

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

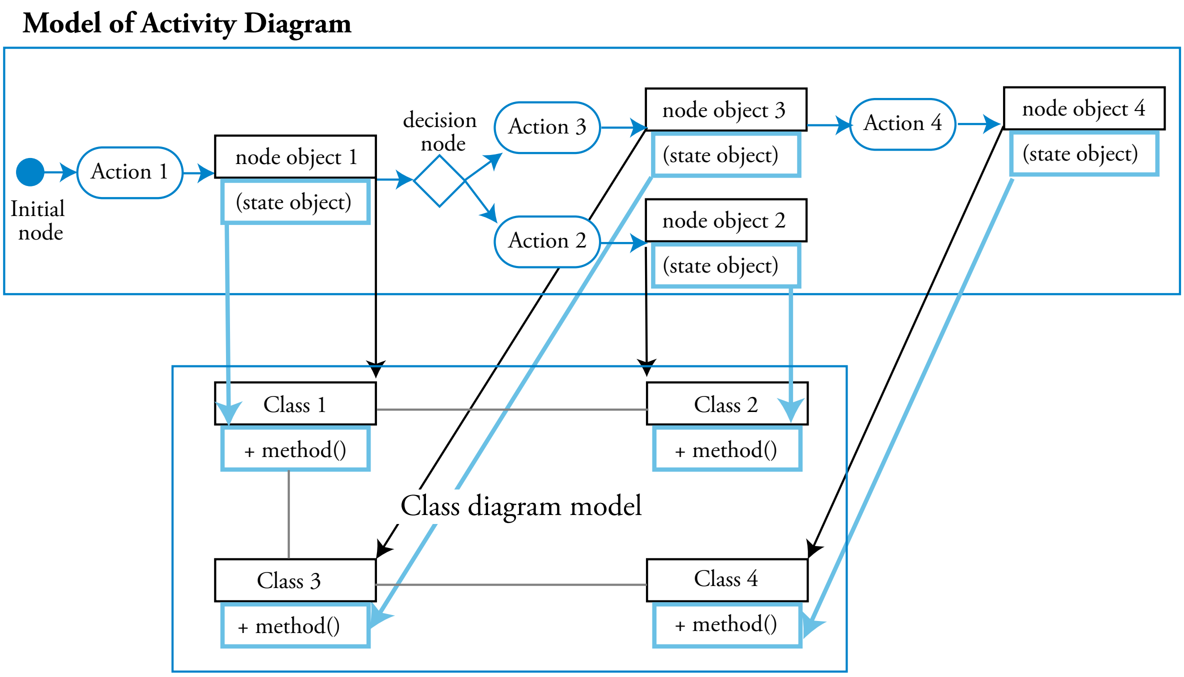

Figure 5. Schema of passage from activity diagram model to class diagram model

In this model, each object node is transformed into class. Then the states of an object node become functions in the class. For example, in hotel accommodation, the object node «reservation» with state «started» is transformed into class «reservation» that contains the method «start».
The rules of passage from the model of Data Flow Diagram and the model of class diagram to the model of package diagram (Figure 6):

- Each process is transformed into a package
- Classes resulting from the same process will be placed in the package that corresponds to the processes.

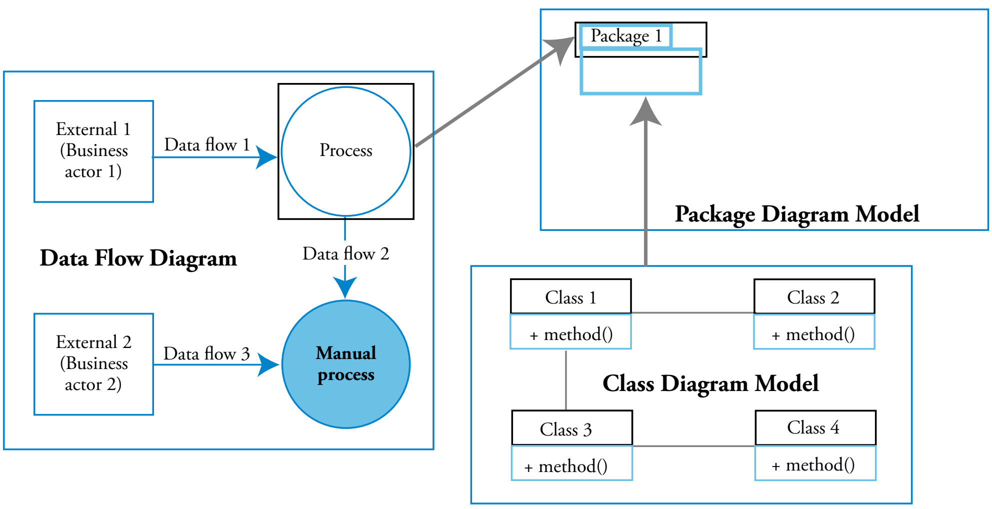

Figure 6. Schema of passage from DFD model and class diagram model to package diagram model

In this model, for example in hotel accommodation, the process such as «deal accommodation» becomes a package. In this section, we presented construction rules for structuring CIM in order to facilitate the transformation toward the PIM. For example, we defined an initial DFD model that represents the business process in a generic way as a set of processes (which may be detailed in 4 to 10 actions) each process will be detailed later in the model of activity diagram as several node objects and actions. Indeed, the object node will be transformed into class and the process will be transformed into package. Thereby the classes are distributed in packages, each package containing between 4 and 10 classes. Then, we presented transformation rules to move from business models, represented in the CIM level, to the analysis and design models, shown on the PIM level. In the next section we represent the use of construction rules and transformation rules in real case in order to substantiate our approach.

3. 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 and 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 satisfy 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

3.1 Presentation of the CIM Level

According to MDA (OMG-MDA, 2014), the CIM level must be represented by business process models. Indeed, we find several standards enabling business process modeling such as BPMN, UML activity diagram, SoaML and DFD. In this paper, we based on UML activity diagram because it is an OMG standard for modeling business process, and on DFD because it is a simple standard since it contains a limited number of notations.

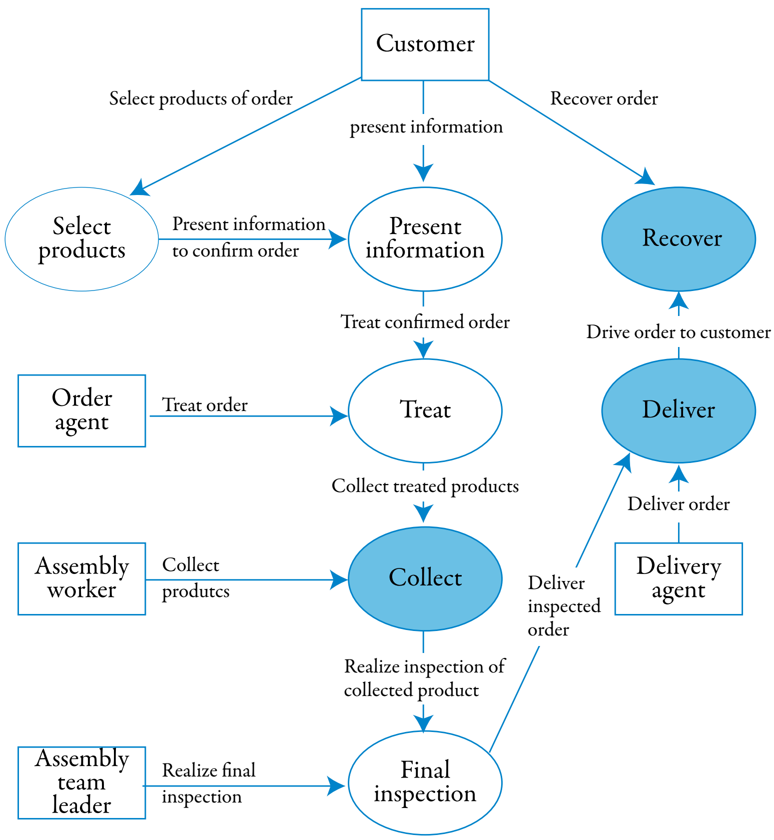

Figure 7. Data flow diagram model of «sales through e-commerce».

Figure 7 shows the business process model represented by Data Flow Diagram. We just specified processes and their sequence to present a business process in general. We tried to present the maximum of business actors to define a true business process, in which there is collaboration between several business actors. For example, instead of putting a single external «delivery service», we identified the externals: «assembly worker», «assembly team leader» and «delivery agent».

Figure 8 shows the second model on CIM level as a model of activity diagram. Through this model we individually detail each process of the previous model as several actions. However, in this model the process «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.

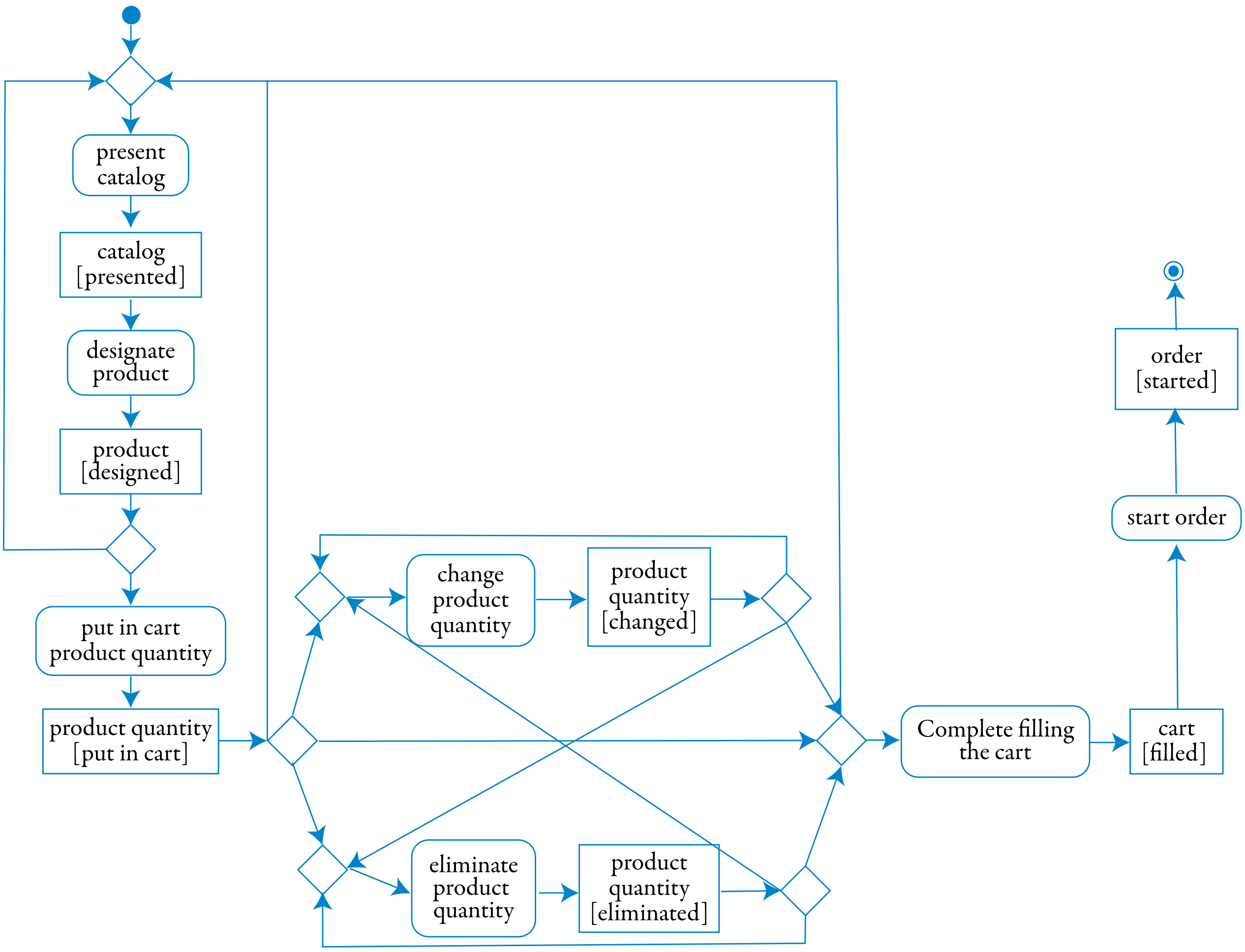

Figure 8. Detailed model of activity diagram of «select products for order"

3.2 Presentation of the PIM level

Figure 9 illustrates a model of use case diagram. In this model, the process «select product for order» -model of DFD- is transformed into a package. Then, the collaborator «customer» who performs the processes becomes actor. Then the actions that detail the processes in the model of the activity diagram are transformed to use cases. Decision nodes that lie between two actions become relationship «extend». However, control flows that lie between two actions become relationship «include."
Figure 10 shows state diagram model transformed from the model of activity diagram of CIM. In this model the states are obtained from nodes of objects. Then, the control flow, which connects two actions, is transformed into a transition. E.g. the object node «catalog» with state «presented» becomes «catalog presented» in state diagram model. Then, initial state is transformed as initial node; final node becomes a final state; node fusion is transformed to junction point; decision node becomes a decision point and fusion node is transformed into a junction point.

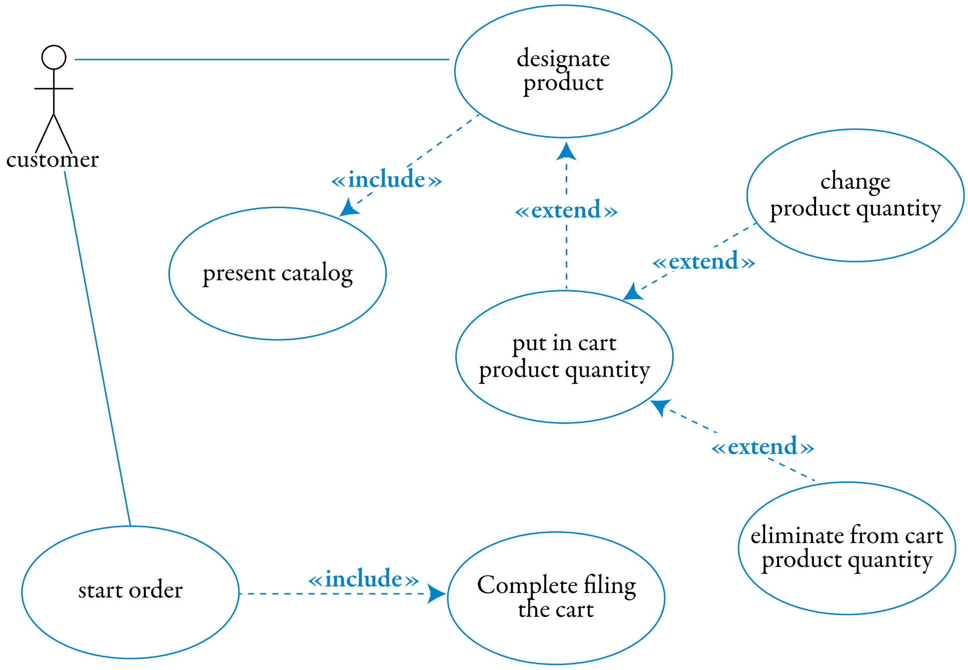

Figure 9. Use case diagram model of «select products of order».

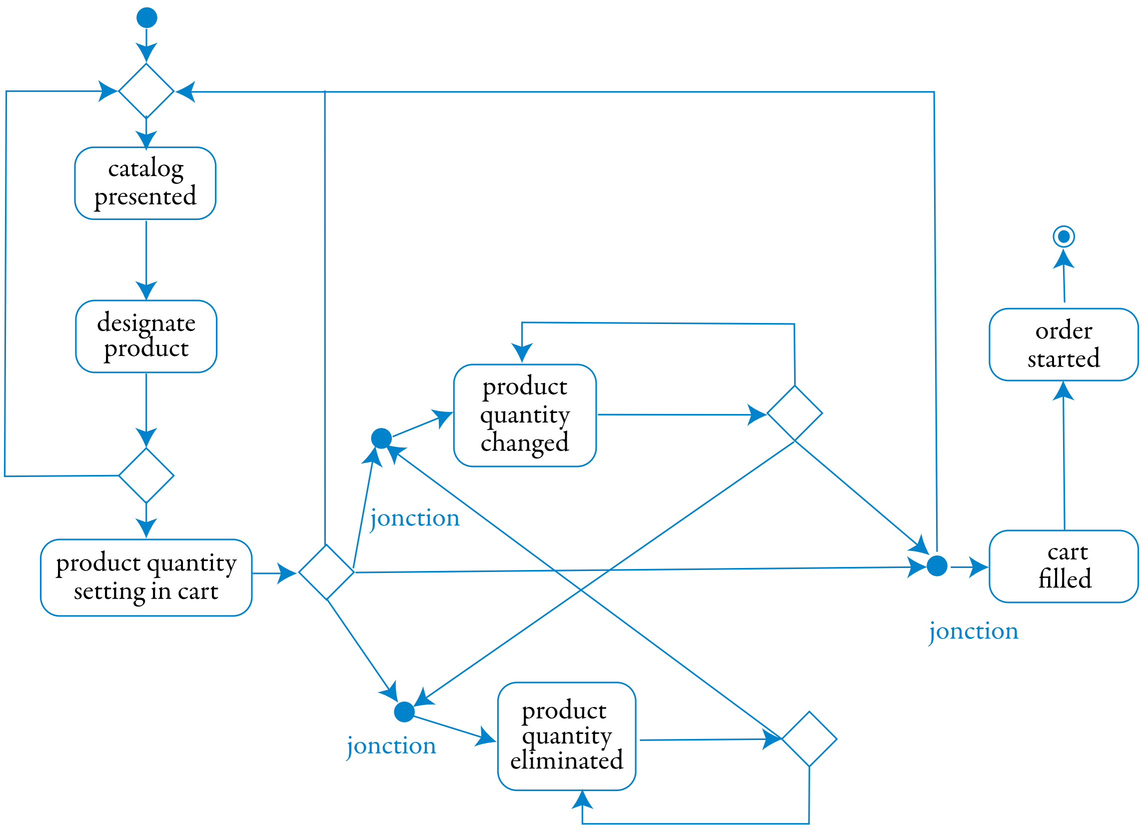

Figure 10. Model of state diagram of «select products of order"

Figure 11 shows the final objective of the PIM level that 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 into functions of the class. So the object node «order» with state «started» is transformed into class «order» that contains the «start» method. Figure 11 shows a model of the package diagram. So, the process «treat order» and «final inspection» become packages.

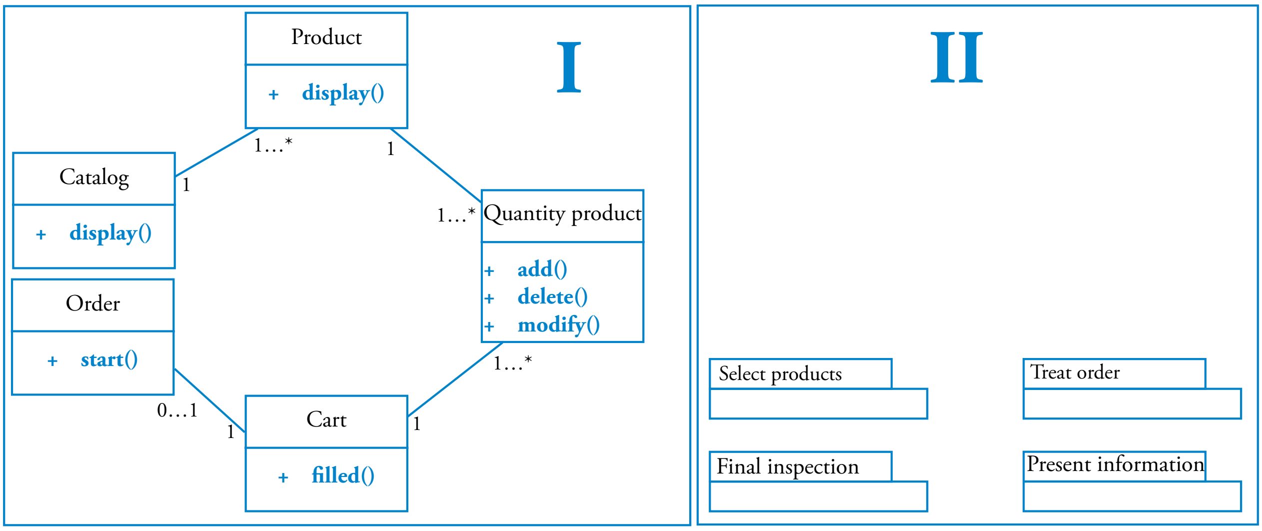

Figure 11. I: model of class diagram of «select products of order», II: model of package diagram of «sales through e-commerce»

This section provides a case study that was presented to validate construction rules of CIM level and transformation rules into the PIM level. In the next section we will present the difference between our proposal and the related works.

4. Related works

MDA does not provide indication about the standards that must be used to model different levels. The key principle of MDA is the use of models in different phases of an application development cycle. Specifically, MDA advocates the establishment of business process models on CIM level, analysis and design models on PIM level and code models on PSM level. Indeed relevant research use business process modeling standards such as BPMN, activity diagram, DFD, to model the CIM level. However most research based on UML diagrams to model PIM level because UML is recommended by MDA on this level.

4.1 Evaluation criteria

Our work is based on (Krouile *et al.*, 2013) to establish valuation criteria. According to (OMG-MDA, 2014), CIM is established by business process models and the PIM level is founded on one or multiple design models. Then, transformation rules allow moving from CIM models to PIM models. According to OMG, it is necessary to model all points of view in order to understand the future system.
The evaluation criteria are CIM business modeling, PIM completeness, transformation elements, and Assessment methodology. The CIM level must show the business process by one or more models of business process standards. The PIM level is complete if it holds one or more models for each modeling view: functional, static and dynamic views. However, the main model is class diagram model, because it contains information system structure and it is easily transformable into PSM.
Model transformations are based on source meta-model, target meta-model and transformation rules.

4.2 Analysis and discussion

On the CIM level, some researches focuses only on system requirement for modeling like in (Gutiérrez *et al.*, 2008). Other hybrid solutions are founded on business process and system requirement for modeling the CIM-level as in (Kherraf *et al.*, 2008). In these approaches, system requirements are early modeled on CIM-level to facilitate transformation towards PIM-level. In our method, we modeled business process on CIM-level by DFD and activity diagram.

On the PIM level, there is no approach that covers the three modeling views except (Kardoš *et al.*, 2010) (Rhazali *et al.*, 2015a) (Rhazali *et al.*, 2015d). Also, several approaches do not model the classes on the PIM level as (Zhang *et al.*, 2005) (Castro *et al.*, 2011) (Hahn *et al.*, 2008) (Mazón *et al.*, 2007) (Gutiérrez *et al.*, 2008), although without classes the source code is not easily obtained by transformation.
Our method covers the three modeling views. Therefore, in static view, we relies on class model, and package model for organizing classes.
All approaches define the transformation rules like in (Kherraf *et al.*, 2008) (Kardoš *et al.*, 2010).
Many standards are used in several approaches for modeling business process including :

- Data Flow Diagram (DFD) (Hoffer *et al.*, 2004) used in (Kardoš and Drozdová, 2010) and (Mokrys, 2012),
- Integration DEFinition (IDEF) (Mayer, 1995),
- XML Process Definition Language (XPDL) applied by (Mokrys, 2012),
- UML 2 Activity Diagram (AD) used in (Rodríguez *et al.* 2010).

Other notations or methodologies for modeling business process are cited by OMG (OMG-BPMN, 2011) including BPMN, Electronic Business using eXtensible Markup Language Business Process Schema Specifications (ebXML BPSS), UML EDOC Business Processes, Activity-Decision Flow (ADF) Diagram, RosettaNet.
We relied on above criteria evaluation, for comparing the CIM to PIM transformation approaches (Figure 12). the static view.

	Business modeling of CIM		Completeness of PIM						Transformation		
			Functional view		*Static view*		*Dynamic view*		*Source meta-model*	*Target meta-model*	*Trans-formation rules*
Studied papers	Attained	Representa-tion	Attained	Representa-tion	Attained	Representa-tion	Attained	Representa-tion			
(Kherraf *et al.*, 2008)	Yes	UML activity diagram			Yes	Component & class diagram					Yes
(Zhang *et al.*, 2005)											Yes
(Kardoš *et al.*, 2010)	Yes	DFD	Yes	UML Use Case & Sequence Diagram	Yes	Domain diagram	Yes	UML Activity diagram			Yes
(Rodríguez *et al.*, 2010)	Yes	BPMN & UML activity diagram	Ye	UML Use Case diagram	Yes	UML class diagram					Yes
(Castro *et al.*, 2011)	Yes	BPMN	Yes	UML Use Case diagram			Yes	UML Activity diagram	Yes	Yes	Yes
(Hahn *et al.*, 2010)	Yes	BPMN									Yes
(Mazón *et al.*, 2007)											Yes
(Gutiérrez *et al.*, 2008)							Yes	UML Activity diagram	Yes	Yes	Yes
(Rhazali *et al.*, 2015a)	Yes	BPMN	Yes	UML Use Case diagram	Yes	UML class diagram					Yes
Rhazali *et al.* 2015b)	Yes	BPMN & UML activity diagram	Yes	UML Use Case diagram	Yes	UML class diagram	Yes	UML state diagram			Yes
Rhazali *et al.* 2015c)	Yes	UML activity diagram			Yes	UML class diagram	Yes	UML state diagram			Yes
Rhazali *et al.* 2016a)	Yes	BPMN	Yes	UML Use Case diagram	Yes	UML class diagram	Yes				Yes
Our proposal	Yes	DFD & acti-vity diagram	Yes	UML Use Case diagram	Yes	UML class & package diagram	Yes	UML state diagram	Yes	Yes	Yes

Figure 12. Comparison of studied papers through Evaluation criteria

We base on tree categories and each category contains some criteria :

- the category «business modeling CIM» contains the criteria «attained», «representation» and «simplicity»,

- the category «completeness of PIM contains the criteria «functional view», «static view» and «dynamic view», and each criterion contains two sub-criteria «attained» and «representation»,
- the category «transformation» contains the criteria: «source meta-model», «target meta-model» and «transformation rules».

The gray cell means that the criterion is not verified by the approach.

The principal key in MDA approach is the models transformation. Indeed, in MDA there are two elementary transformation kinds: CIM (computing independent model) to PIM (platform independent model) transformation and PIM to PSM transformation. However, most researches propose approaches transforming PIM to PSM (platform specific model), since there are multiple points in common between PIM level and PSM level. Nevertheless, transforming CIM level into PIM level is rarely addressed in research because these two levels are mainly different. All related works propose approaches to control model transformation from CIM to PIM according to MDA. However, our methodology is obtained from an analytical survey. Indeed, from the beginning, in CIM, we consider that we build business process models, which will be automatically transformed into PIM models. Our methodology is based on creating good CIM models, through well-defined construction rules, to facilitate transformation toward the PIM models. So, we establish a rich PIM level, considering the three classical modeling views : dynamic, functional and static. Use case diagram model interprets functional view, state machine diagram model represents the dynamic view, class and package diagram models show

5. Conclusions and future works

One of the main challenges in the software development process is the establishing of an approach that allows moving from models that define the business process to models which represent the analysis and design of software. Based on MDA, our approach provides a solution to the problem of transformation from business models represented on CIM level to design models represented on PIM level. This methodology results in a set of well-structured and useful classes in the process of software development. In this approach, we benefited from our experience in old transformation methods (Rhazali *et al.*, 2015a) (Rhazali *et al.*, 2015d) (Rhazali *et al.*, 2016a) (Rhazali *et al.*, 2015b) to provide, through semi-automatic transformation, a set of classes structured in packages from business models defined by DFD and activity diagram.

Our approach does not transform all notations existing in business process models into PIM level models such as the representation of the fusion node in the use case diagram, which is not considered by our work. Furthermore, several important elements on the PIM level are not automatically transformed from the CIM level like class properties and class relationships. Nethertheless, our approach provided a transformation methodology with some significant transformation rules for solving the transformation problem from CIM level to PIM level.

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

6. References

Blanc, X. (2005). *MDA in action*. Ed. Eyrolles.

De Castro, V., Marcos, E., & Vara, J.M. (2011). *Applying CIM-to-PIM model transformations for the service-oriented development of information systems*. Journal of Information and Software Technology, 53 (1), 87-105.

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

Grammel, B., & S. Kastenholz, A. (2010). *Generic traceability framework for facet-based traceability data extraction in model-driven software development.* in Proceedings, 6th ECMFA Traceability Workshop held in conjunction ECMFA. (pp. 7-14). Paris, France.

Gutiérrez, J.J., Nebut, C., Escalona, M.J., Mejías, M., Ramos, I.M. (2008). *Visualization of use cases through automatically generated activity diagrams*. In Proceedings of the 11th International Conference MoDELS'08, Toulouse, France.

Hoffer, J.A, George, J.F, Valacich, J.S. (2004). *Modern system analysis and design*. Prentice Hall ISBN 0-13-145461-7, 2004.

Kardoš, M., Drozdová, M. (2010). *Analytical method of CIM to PIM transformation in Model Driven Architecture (MDA)*, Journal of information and organizational sciences, vol. 34, pp. 89-99.

Kherraf, S., Lefebvre, É., Suryn, W., (2011). *Méthodologie de transformation du CIM en PIM dans l'approche MDA*. Thèse de doctorat électronique, Montréal, École de technologie supérieure.

Kherraf, S., Lefebvre, É., Suryn, W. (2008). *Transformation from CIM to PIM using patterns and Archetypes*. In ASWEC'08, 19th Australian Software Engineering Conference, Perth, Australia.

Kriouile, A., Gadi, T., Balouki, Y. (2013). *CIM to PIM Transformation: A criteria Based Evaluation*. International Journal Computer Technology & Applications, 4(4), 616-625.

Mayer, R., Menzel, C., Painter, M., Perakath, B., de Witte P. and Blinn T. (1995). *Information Integration For Concurrent Engineering (IICE) - IDEF3 Process Description Capture Method Report*. Technical Report September 1995. available at http://www.idef.com/pdf/idef3_fn.pdf

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: Foundations and Applications, ER Workshops, Auckland, New Zealand, pp. 255–264.

Mokrys, M. (2012). *Possible transformation from Process Model to IS Design Model*. In First International Virtual Conference Slovakia, pp. 71–74.

OMG-BPMN. (2011). *Business Process Model and Notation (BPMN)-Version 2.0*. Boston, USA: OMG.

OMG-MDA. (2014). *Object Management Group Model Driven Architecture (MDA) MDA Guide rev. 2.0.* Boston, USA: OMG.

OMG-QVT. (2015). *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, V1.2*. Boston, USA: OMG.

OMG-SoaML (2012). *Service Oriented Architecture Modeling Language (SoaML) – Specification for the UML Profile and Metamodel for Services (UPMS)*. OMG document: ad/2012-05-10. Available at <`http://www.omg.org/spec/SoaML/1.0.1/PDF`>.

OMG-UML. (2011). *OMG Unified Modeling LanguageTM (OMG-UML)*, Infrastructure, `http://www.omg.org /spec /UML/2.4.1/Infrastructure`. August 2011.

Osis, J., Asnina, E., & Grave, A.. (2007). *Formal Computation Independent Model of the Problem Domain within the MDA*. ISIM.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2015a). *A Methodology of Model Transformation in MDA: from CIM to PIM*. International Review on Computers and Software, 10 (12), 1186-1201. DOI: http://dx.doi.org/10.15866/irecos.v10i12.8088.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2015b). *A Methodology for Transforming CIM to PIM through UML: From Business View to Information System View*. In Proceedings Third World Conference on Complex Systems. Marrakech, Morocco. DOI: 10.1109/ICoCS.2015.7483318.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2015c). *Disciplined Approach for Transformation CIM to PIM in MDA.* In Proceedings, 3rd International Conference on Model-Driven Engineering and Software Development. (pp. 312 – 320). Angers, France.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2015d). *Transformation Approach CIM to PIM: From Business Processes Models to State Machine and Package Models*. In Proceedings, the 1st International Conference on Open Source Software Computing. (pp. 1 – 6). Amman, Jordan. DOI: 10.1109/OSSCOM.2015.7372686.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2016a). *A Based-Rule Method to Transform CIM to PIM into MDA.* International Journal of Cloud Applications and Computing, 6 (2). DOI: 10.4018/IJCAC.2016040102.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2016b). *CIM to PIM Transformation in MDA: from Service-Oriented Business Models to Web-Based Design Models*. International Journal of Software Engineering and Its Applications, 10 (4), 125-142. DOI: 10.14257/ijseia.2016.10.4.13.

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2016c). *A New Methodology CIM to PIM Transformation Resulting from an Analytical Survey*. In Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development. (pp. 266-273). Rome, Italy. DOI: 10.5220/0005690102660273

Rhazali, Y., Hadi, Y., & Mouloudi, A. (2016d). *Model Transformation with ATL into MDA from CIM to PIM Structured through MVC*. Procedia Computer Science - Journal - Elsevier. doi:10.1016/j.procs.2016.04.229.

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.

Roques, P., 2004, *UML in Practice: The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions*. Wiley.

Schmidt, D.C. (2006). *Guest Editor's Introduction: Model-Driven Engineering*. IEEE Computer, 39 (2), 25 - 31.

Zhang,W., Mei, H., Zhao, H., & Yang, J. (2005). *Transformation from CIM to PIM: A Feature-Oriented Component-Based approach*. In Proceedings MoDELS. (pp. 248-263). Montego Bay, Jamaica.