*Journal of Ubiquitous Systems & Pervasive Networks*
*Volume 8, No. 1 (2017) pp. 19-26*

# A Model transformation from computing independent model to platform independent model in model driven architecture

**Yassine Rhazali*, Youssef Hadi, Abdelaziz Mouloudi**

Ibn tofail University, Kenitra, Morocco

## Abstract

This paper presents a new methodology to master models transformation from CIM to PIM in accord with MDA approach. Our improved approach is founded on creation a good CIM level, through well-chosen rules, for facilitating models transformation to PIM level. However, we establish a rich PIM models via use case diagram, state diagram, class and package diagram. Next, we describe a set of transformation rules to ensure a semi-automatic transformation from CIM to PIM. Our method conform MDA approach by taking into consideration the business dimension in the CIM level, since, we present this level by using activity diagram who is a standard of business process modeling. Nevertheless, we use UML into PIM in our proposal, because UML is recommended by MDA in this level.

***Keywords:*** *MDA, computer model, CIM, PIM, business process modeling, model transformation, computer modeling, business models*

## 1. Introduction

Transformations between different levels of MDA [1] begin by CIM-to-PIM transformation, which allows building PIM models from CIM models. The goal is to reword information contained in CIM models inside PIM models, which ensures that business information are not disappeared throughout MDE[4] process. Then, transformation from PIM models toward PSM models allows adding in PIM a set of technical information related to a target platform.

In practice, automatic transformation begins from PIM to PSM. However, our ultimate aim is to make the CIM a productive level, and a basis for building PIM through an automatic transformation. The goal is that business models do not remain only simple documents of communication between business experts and software designers.

In this paper, we present a solution for automating the transformation of CIM level to the PIM level. However, we use UML 2 activity diagram in an efficient manner so as to attain concentrated CIM models, which simplifies the transformation towards PIM level. Then, we define a set of well selected-rules for automating the transformation from CIM to PIM.

Our approach is based on UML 2 [2] activity diagram ,which represents one of standards of business model, for define the CIM level. Then rich business models of well-concentrated informations help us to obtain easily models of PIM level. First model of PIM level is use case diagram model allows modelling functinalities of the system.Then,the state diagram model defines the states of information system. Next, class diagram model allows modeling system classes and their relationships independently of a programming language. Finally, all classes are organized inside packages into package diagram model.

Rest of this paper is presented as follows. In section II we analyze the related works concerning transformation from CIM to PIM. Section III presents our proposal and describes constructing rules of CIM models and transformation rules allow shifting from CIM models to PIM models. In Section IV we illustrate our method in a case study demonstrating the construction of the CIM level and its transformation to PIM level. Finally, Section V, we conclude by specifying outcome of our work and determining future works.

## 2. Related Work

In this section, we present related works concerning transformation from CIM level to PIM level in MDA, describing advantages and disadvantages of each method.

A transformation from CIM to PIM oriented service was presented by Castro et al. [6]. Authors represent CIM level by using BPMN [3] for modeling business process, and by using value model [5] for identifying services from the beginning in the business perspective. Authors use ATL language, for moving towards PIM level; however, this level is presented by two extensions of use case model and two extensions of the activity diagram. This approach has the advantage of identifying services and business process at CIM level in order

* Corresponding author. Tel.: +212666525892
E-mail: dr.yassine.rhazali@gmail.com

DOI: 10.5383/JUSPN.08.01.003

to guide a semi-automatic transformation to PIM level. But authors study is only limited to use case diagram and activity diagram in PIM level and does not present the structural view (generally represented by the class diagram) that defines the ultimate objective of this level. Nevertheless, the use of activity diagram in the PIM level represent a great inconvenience because this diagram is considered one of standards for modeling business processes.

A transformation method from CIM to PIM, based on security requirements to represent business perspective in CIM is presented by Rodríguez et al. [7]. The authors is based on BPMN notation for modeling business processes into CIM level; then, they use QVT [12] language to transform CIM in order to obtain class diagrams and use case in PIM. This approach shows very interesting ideas for transforming CIM to PIM into security-oriented field. However, this method based only on secure information systems.

Hahn et al. [9] is based on engineering services driven by models. The authors represent CIM level through BPMN, then, they use ATL language to move to PIM level represented in this method by SoaML models. This approach apply SoaML, the new OMG standard to model services, but this method does not represent the ultimate aim of PIM level that is presented in use of one or more structural diagrams (as class diagram).

Zhang et al. [10] describe a method in that CIM and PIM are respectively represented by features and components. Responsibilities in this approach are considered as connectors between features and components to facilitate transformation between CIM and PIM. Grammel and Kastenholz [11] rely on a DSL connection for manage traceability in general. Both methods offer solutions for transforming CIM level to PIM level, while they do not define models used in CIM level and PIM level.

An approach according MDA, its goal is transforming use case diagram to activity diagram is represented by Gutiérrez et al. [13]. The authors founded on QVT to transform existing use cases to activity diagram. While this method allows to transform CIM to PIM via clear rules, the authors specify in CIM level the functional requirements represented by the use case.

Mazon et al. [14] present an objective-oriented method by defining a UML profile to establish the CIM level, founded on i* modeling framework. The authors based on QVT for moving to PIM, which focuses on conceptual modeling of data warehouse. However, this method deals the transformation from CIM to PIM only in the field of data warehousing.

Kherraf et al. [8] define a method for transforming CIM to PIM. The authors use 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 represents the last step in the CIM level. The system requirements are transformed as components of the system. These are shown in the component diagram as a first stage in the PIM. Finally, a set of business archetypes helps to transform the components of system to class diagram. This method proffers interesting ideas allow to facilitate the transformation from CIM to PIM. Nevertheless, this approach uses diagram of use case in the CIM level even if it represents the system functionalities.

Kardos et al. [15] propose an analytical method for transforming CIM to PIM in MDA. The authors use data flow diagram in the CIM level; then they base on the use case diagram for introduce the information system view. This method defines a model of activity diagram as well as a model of sequence diagram and a class diagram model. The benefit of this approach is the use of several UML diagrams that present different views of the information system in PIM, but this approach does not represent a real business view since it based on DFD in CIM level, and does not clearly describe transformation rules from CIM to PIM.

A transformation method from business process models to use case and class diagram is proposed by Rhazali et al. [16]. The authors use BPMN to model business process in CIM level. Nevertheless, UML 2 use case diagram and class diagram present PIM level. We highlight in our related work an interesting research about model transformation in MDA proposed by Drozdova et al.[30].

After this overview on related works about transformation from CIM level to PIM level, we can divide the works into five types. We note approaches [8], [13] that use model of system requirements (as use case diagram) early in the CIM level, to facilitate transformation to PIM level. However, other works [6], [9] even if they base on business processes in CIM level, do not establish the structural view (usually through the class diagram) in PIM level. Nonetheless, there are methods which aim transformation in a particular area [7], [14]. Then, there are researches like in [15] that establish the structural view in the PIM level and are not oriented to a particular field, but the authors do not specify rules of transformation. Finally, there are approaches [10], [11] which describe precisely the transformation rules, but do not have the models used in the CIM and PIM. In [16], [17] the authors propose interesting approach concerning transformation from CIM level to PIM level, but it is based on BPMN to create the CIM level for this the authors are limited when transforming the CIM level into the PIM level.

## 3. Proposed Method of Transformation from CIM to PIM

A business model describes abstractly how the business is working. According to our modeling needs we can, more or less, make different models to describe the same reality. However, we can model business process to improve communication with customers or partners, for controlling business process, or to establish an information system. In this paper, our object is to design business process models as a first stage in the process of information system development. In our case, we have two alternatives: either create business models in the form of documents that will be transformed manually via analysts and software designers, or to create business models that will be transformed in an automatic way. The second choice was regarded in our method for designing effective business models that contain rich information to facilitate the transformation. However, we respect MDA approach in the conception and transformation of business models.

Our study considers business dimension into the presentation of CIM level, through the use of real business model, to preserve the knowledge of the business throughout the transformation towards PIM level. This enables us the realization of quality information system.

UML 2 activity diagram is one of standards of business modeling. In our method, we use effectively UML 2 activity diagram to present two extensions of business processes models for achieving a concentrated CIM level that help us to obtain readily models of PIM level.

MDA recommends the use of UML in the PIM level. The use case presents functional view. Then, the state diagram interprets the dynamic view and the class diagram shows the static view of the information system, finally we structure all classes in packages through package diagram (cf. Fig. 1)

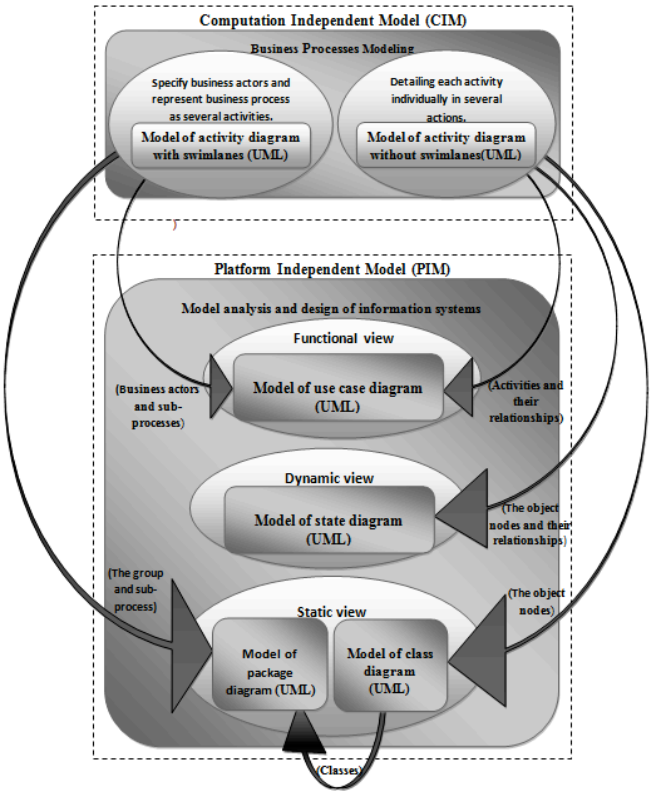


**Fig 1. Schema of the proposed approach**

All PIM models obtained via an automatic or semi-automatic transformation from CIM level. Transformation is realized through concentrated rules.

### 3.1. Construction Rules of CIM Level

The rules for constructing the general model of activity diagram with swimlanes (cf. Fig. 2):

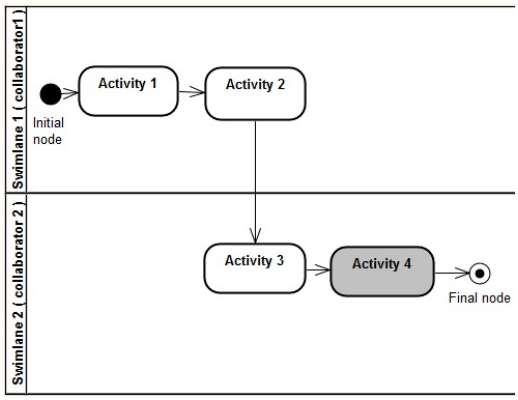


**Fig 2. Generic model of UML activity diagram (with swimlanes).**

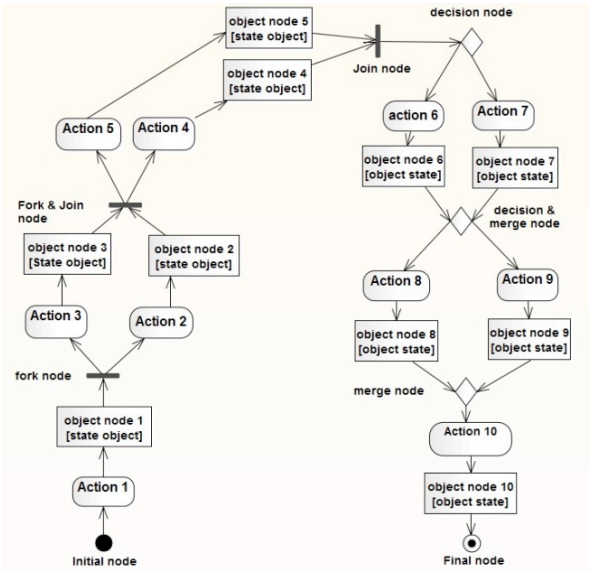


**Figure 3. Geniric model of UML activity diagram (without swimlanes).**

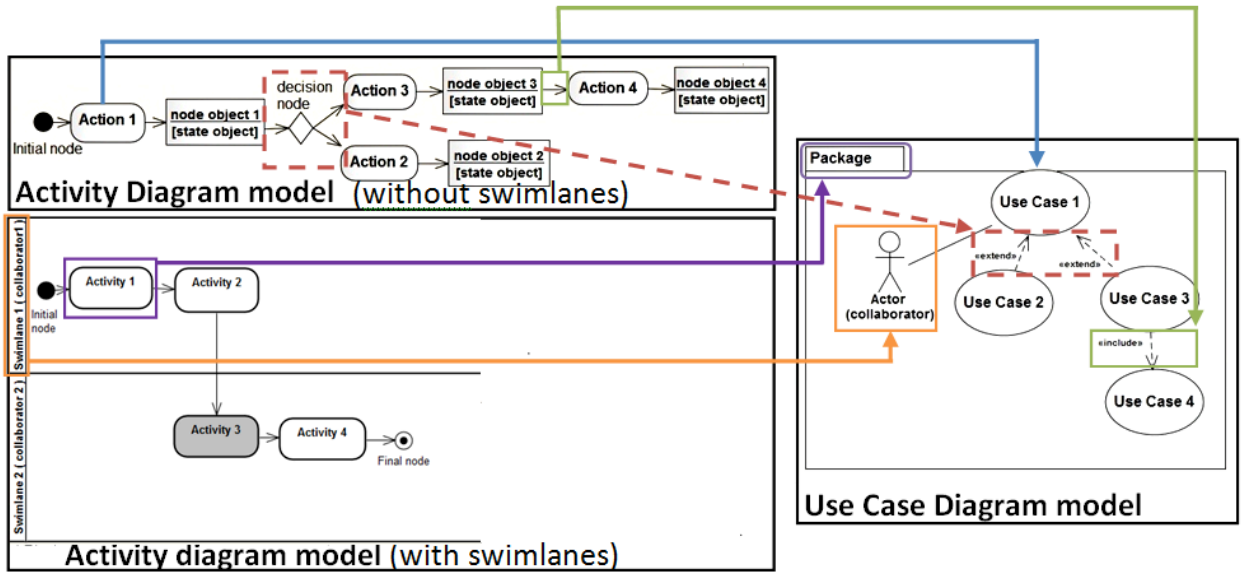


**Fig 4. Schema of passage from activity diagram models (with and without swimlanes) to use case diagram model**

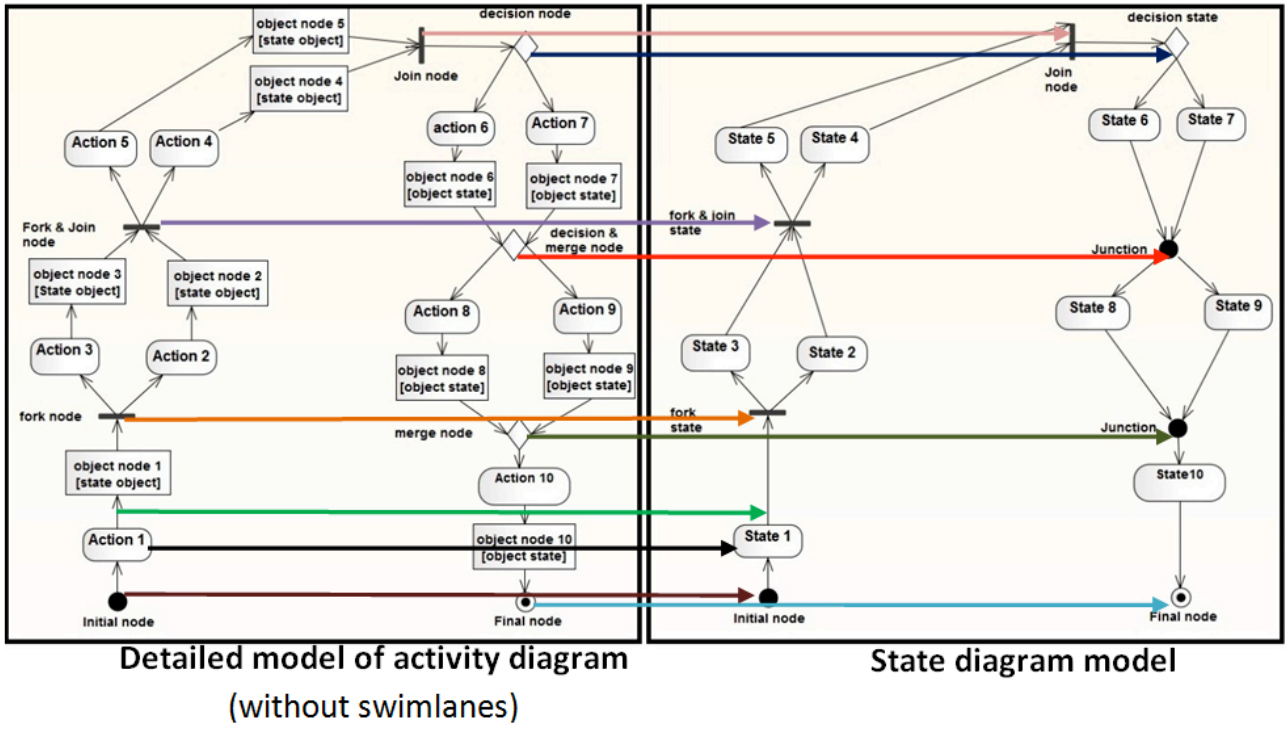


**Fig. 5. Schema of passage from detailed model of activity diagram (without swimlanes) to class diagram model**

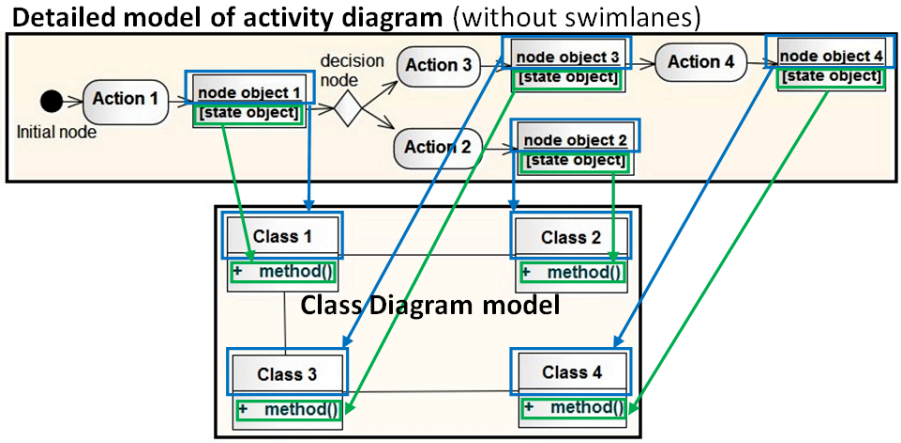


**Fig. 6. Schema of passage from detailed model of activity diagram (without swimlanes) to class diagram model**

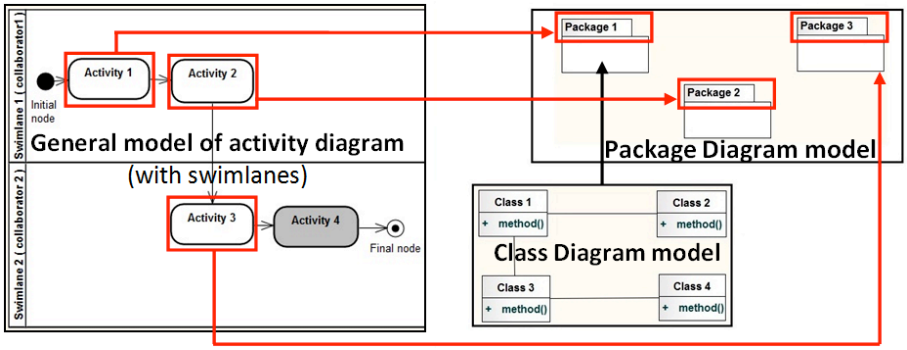


**Fig. 7. Schema of passage from general model of activity diagram (with swimlanes) and class diagram model to package diagram model**

- Define averages activities (not complexes activities). In fact, each activity must be comprised between 4 and 10 actions.
- If an activity consists of less than 4 actions, or represents a complementary operation to another activity, we can merge several activities into one. Provided that the activity do not exceed 10 actions.
- Avoid the maximum possible, the representation of actions.
- The model does not describe all cases and paths, but it presents just a description of the sequence of activities of the most common business processes.
- Focus on activities and their sequences.
- Coloring manual activities with another color for example we used the gray
- Use the "Region" notation to group activities that belong to the same category
- Identify the maximum possible of the actors who interact and who collaborate in the achievement of business processes since we are talking about an enterprise process.
- Avoid in this model, the maximum possible, representation of the gateways.

The rules of construction of detailed model of the activity diagram without swimlanes (cf. Fig. 3):

- Detail individually each activity in a model as a several actions (this latter constitute the fundamental unit in the activity diagram).
- Do not present, into this model, manual tasks of general model of activity diagram (with swimlanes).
- Show connections in this model.
- Representing in this model of the most outstanding ways.
- Any action described in model of activity diagram (with swimlanes) will be shown here by an action.
- Add an object node, which contains object state, at the output of each action.

### 3.2. Transformation Rules from CIM to PIM

The rules of passage from models of the activity diagram (with and without swimlanes) to model of use case diagram (cf. Figure 4):

- Every action of detailed model of activity diagram that corresponds to a functionality of the system is transformed to use case.
- The collaborator, who realizes the activity of general model of activity diagram, becomes an actor use cases that correspond to the actions of this activity.
- If there is "decision node" between two actions, the corresponding use cases connect by a relationship "extend".
- Do not transform the control flow returning back
- there was just a control flow between two actions, the corresponding use cases connect by a relationship "include".
- Each activity of general model of activity diagram is transformed to a package which includes the use cases corresponding to the actions of this activity

The rules of passage from detailed model of the activity diagram (without swimlanes) to model of state diagram (cf. Fig. 5):

- An object node transforms to a state.
- A decision node transforms to a decision point.
- A merge node transforms to a junction point.
- A decision and merge node transforms to a junction point.
- An initial node transforms to an initial state.
- A final node transforms to a final state.
- A control flow between two actions transforms to a transition.
- A fork node transforms to a fork state.
- A joint node transforms to a joint state.
- A joint and fork node transforms to a joint and fork state.

The rules of passage from detailed model of the activity diagram to the model of class diagram (cf. Fig. 6):

- An object node transforms to class.
- A state of an object node transforms to a class method.

The rules of passage from general model of activity diagram (with swimlanes) and the model of class diagram to the model of package diagram (cf. Fig. 7):

- A region transforms to a package.
- An activity which does not belong to any region becomes a package.
- A set class, which becomes the same region, will be placed in the package corresponding to the region.
- Classes coming from the same activity, that does not located in any region, will be placed in the package matching to the activity.

## 4. Case Study

In this section, we show a case study of sales through e-commerce to illustrate our method of transforming CIM level to PIM level.

A costumer can show items available in the catalog. He can also see detailed information concerning each product, then he decides to put a quantity of product in cart or not. In any moment the customer has the right to modify the quantity or eliminate entirely the product from the cart. Once products that match to the needs are definitely selected by the customer, the latter can start the command. Then, he shows payment information, and chooses delivery details.
An order agent treats the order, and declares the reservation of products chosen by the customer. Then, the assembly worker collects manually the reserved articles from stock.
For each product the assembly team leader inspects the quality and quantity. Then the delivery agent delivers the confirmed order, in order that customer obtains his products.

### 4.1 Presentation of the CIM Level

Fig.8 shows first model in CIM level, through business process established by our general model of activity diagram (with

swimlanes). In this model we just specify the activities and their sequencing, by avoiding the identification of actions and gateways for presenting business process in general.

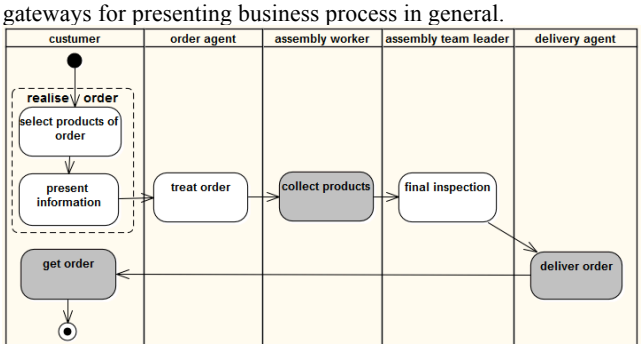


**Fig. 8. model of activity diagram (with swimlanes) of "sales through e-commerce"**

Fig. 9 shows the second model in CIM level as a detailed model of activity diagram (without swimlanes). In this model we detail individually each activity of the previous model as a several actions. However, in this model the activity "select product for order" is analyzed. Then, we have presented all possible paths. Then we identify an object node together with its state in the output of each action.

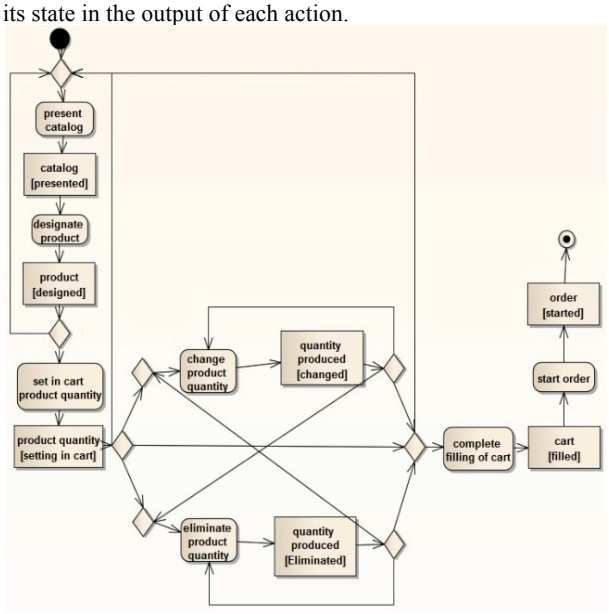


**Fig. 9. model of activity diagram (without swimlanes) of "select products of order".**

### 4.2 Presentation of the PIM level

Figure 10 presents a model of diagram use case. This model is transformed from the business models of CIM level. However, in this model the activity "select product for order" is transformed to a package. Then, the collaborator "customer" who performs the activity becomes actor. Then the actions that detail the activity 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.

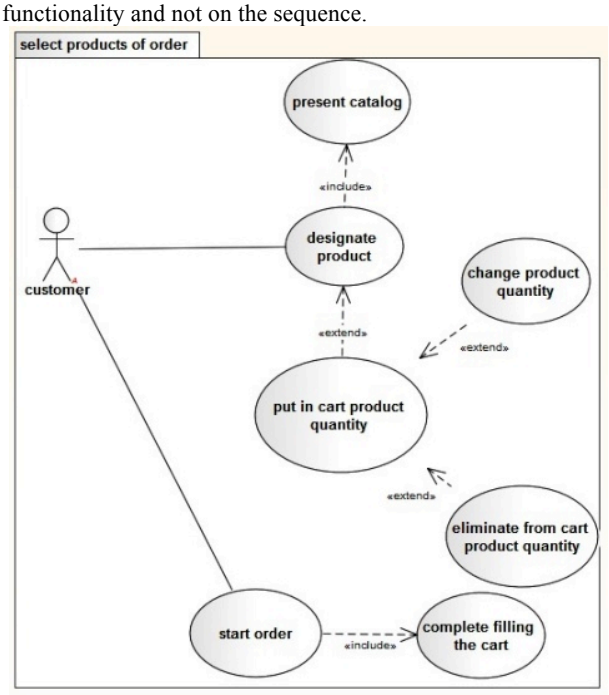


**Fig. 10. Use case digram model of "select products of order".**

Fig. 11 shows state diagram model transformed from the detailed 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 to a transition. E.g. the object node "catalog" with state "presented" becomes "catalog presented" in state diagram model. Then, initial state is transformed from initial node; final node becomes to a final state; node fusion transformed to junction point; decision node becomes a decision point and decision and fusion node transformed to a junction point.

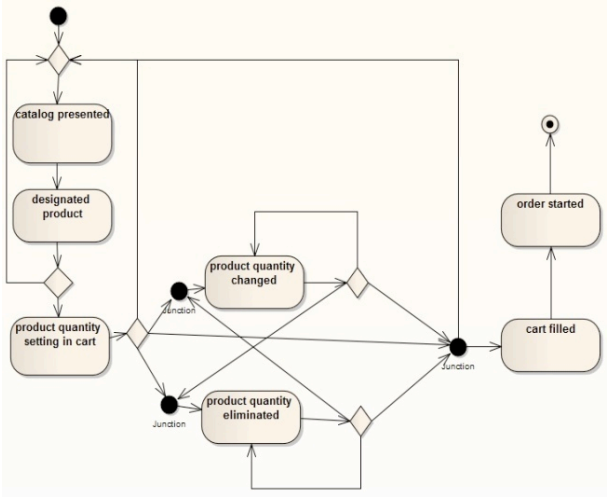


**Fig. 11. State digram model of "select products of order".**

Fig. 12 represents class diagram model, which is the final goal of the PIM level. This model is obtained from the detailed model of activity diagram. In this model object nodes are transformed to classes. Then functions of the class are

transformed from states of an object. E.g. the object node "order" with state "started" transform to class "order" that holds the "start" method.

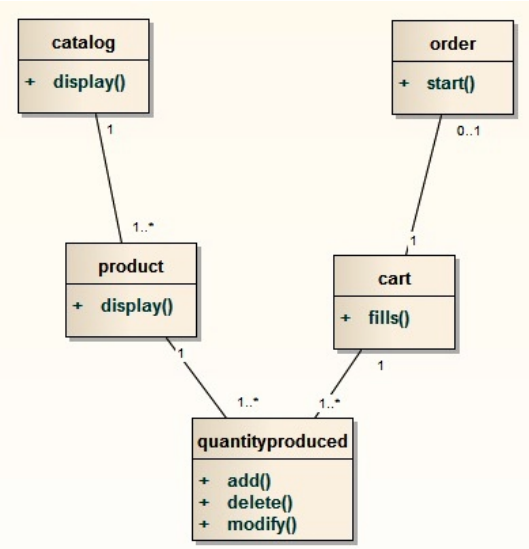


**Fig 12. Class diagram model of "select products of order".**

Fig. 13 represents model of package diagram. So the region "realize order" becomes package. Then the activities that are not in a region, such as "treat order" and "final inspection" are transformed to packages.

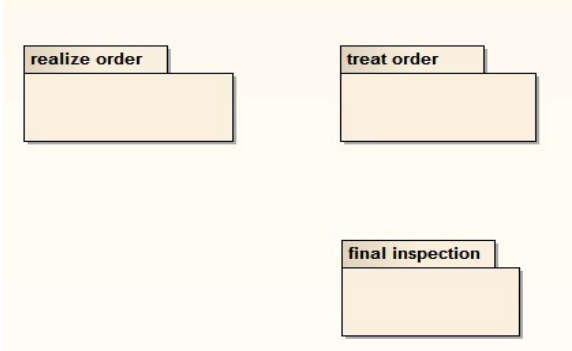


**Fig13. package diagram model of "select products of order".**

## 5. Analysis And Discussion

Our actual method present a new improved approach from old approaches [16], [17], [18], [19], [20], [21] , [22] , [23] , [24] , [25] , [26] , [27] , [28] , [29]. 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. For example with BPMN we cannot present object state in CIM models etc...

We find that the approach [6] validates most of our criteria. This approach allows building the CIM level on the basis on value model that is not a business modeling standard. Then the PIM level is presented just by the models of the use case 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 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 and Future Work

One of the major challenges in software development process is the establishment of a methodology that allows shifting from models that present the running of the business to models which describe the analysis and design of software.

Founded on MDA, our methodology provides a solution to the difficulty of transformation from business models, represented in CIM level, to the analysis and design models, modeled in PIM level. This method achieves a set of well organized and useful classes in the process of software development.

The ongoing work is intended to implement, construction rules and transformation rules, in a tool through the ATL language. However, in our future work, we anticipate to transform the models obtained in the PIM level to models of PSM level, indeed, our ultimate objective is to provide the source code from the business models by means of automatic transformations, By inspiring from [31] and [32].

## References

[1] OMG-MDA, "MDA Guide Version 1.0.1," 1 juin 2003.

[2] OMG, UML Superstructure 2.0. OMG Adopted Specification ptc/03-08-02, 2003. <http://www.uml.org/>.

[3] OMG, "Business Process Modelling Notation", Version 2.0, 2011.

[4] C. Schmidt, "COVER FEATURE Model Driven Engineering", 2006.

[5] J. Gordijn, and J. M. Akkermans, "Value based requirements engineering: exploring innovative e-commerce idea," Requirements Engineering Journal 8 (2), 2003, pp. 114–134.
https://doi.org/10.1007/s00766-003-0169-x

[6] V. D. Castro, E. Marcos, and J. M. Vara, "Applying CIM-to-PIM model transformations for the service-oriented development of information systems," presented at 2nd Information and Software Technology, 2011, pp. 87–105.
https://doi.org/10.1016/j.infsof.2010.09.002

[7] A. Rodríguez, I. García-Rodríguez de Guzmán, E. Fernández Medina, and M. Piattini, "Semi-formal transformation of secure business processes into analysis class and use case models: an MDA approach," presented at 9th Information and Software Technology 52, 2010, pp. 945–971.
https://doi.org/10.1016/j.infsof.2010.03.015

[8] S. Kherraf, E. Lefebvre, and W. Suryn, "Transformation from CIM to PIM using patterns and archetypes," presented at 19th Australian Conference on Software Engineering, 2008,pp. 338-346.
https://doi.org/10.1109/aswec.2008.4483222

[9] C. Hahn, P. Dmytro, and K. Fischer, "A model-driven approach to close the gap between business requirements and agent-based execution," presented at Proceedings of the 4th Workshop on Agent-based Technologies and

applications for enterprise interoperability, Toronto, Canada, 2010, pp. 13–24.

[10] W. Zhang, H. Mei, H. Zhao, and J. Yang, "Transformation from CIM to PIM: a feature-oriented component-based approach," presented at MoDELS 2005, Montego Bay, Jamaica, 2005.
https://doi.org/10.1145/1814392.1814394

[11] B. Grammel, and S. Kastenholz, "A generic traceability framework for facet-based traceability data extraction in model-driven software development," presented at the 6th ECMFA Traceability Workshop held in conjunction ECMFA 2010, Paris, France, 2010, pp. 7–14.

[12] OMG, "MOF 2.0 Query/View/Transformation (QVT)", V1.0. OMG Document – formal/08-04-03, 2008.

[13] J. J. Gutiérrez, C. Nebut, M. J. Escalona, M. Mejías, and I. M. Ramos, "Visualization of use cases through automatically generated activity diagrams," presented at 11th international conference on Model Driven Engineering Languages and Systems, France, 2008, pp. 83-96.
https://doi.org/10.1007/978-3-540-87875-9_6

[14] J. Mazón, J. Pardillo, and J. Trujillo, "A model-driven goal-oriented requirement engineering approach for data warehouses," presented at the Conference on Advances in Conceptual Modeling: Foundations and Applications, Auckland, New Zealand, 2007, pp. 255–264.
https://doi.org/10.1007/978-3-540-76292-8_31

[15] M. Kardoš, and M. Drozdová, "Analytical method of CIM to PIM transformation in model driven architecture (MDA)," : JIOS, VOL. 34, NO. 1, 2010, pp. 89-99.

[16] Y. Rhazali, Y. Hadi, A. Mouloudi, Transformation Method CIM to PIM: From Business Processes Models Defined in BPMN to Use Case and Class Models Defined in UML, International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 8, n.8, pp. 1453–1457, 2014.

[17] Y. Rhazali, Y. Hadi and A. Mouloudi, "Disciplined approach for transformation CIM to PIM in MDA," Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference on, Angers, 2015, pp. 312-320.
https://doi.org/10.5220/0005245903120320

[18] Y. Rhazali, Y. Hadi and A. Mouloudi, "A methodology for transforming CIM to PIM through UML: From business view to information system view," 2015 Third World Conference on Complex Systems (WCCS), Marrakech, Morocco, 2015, pp. 1-6. https://doi.org/10.1109/ICoCS.2015.7483318

[19] Y. Rhazali, Y. Hadi and A. Mouloudi, "Transformation approach CIM to PIM: from business processes models to state machine and package models," Open Source Software Computing (OSSCOM), 2015 International Conference on, Amman, 2015, pp. 1-6.
https://doi.org/10.1109/OSSCOM.2015.7372686

[20] Y. Rhazali, Y. Hadi and A. Mouloudi, (2016). Model Transformation with ATL into MDA from CIM to PIM Structured through MVC. Procedia Computer Science, 83, 1096-1101.
https://doi.org/10.1016/j.procs.2016.04.229

[21] Y. Rhazali, Y. Hadi and A. Mouloudi, (2016). CIM to PIM Transformation in MDA: from Service-Oriented Business Models to Web-Based Design Models. International Journal of Software Engineering and Its Applications. Vol. 10, no. 4, pp. 125-142.
https://doi.org/10.14257/ijseia.2016.10.4.13

[22] Y. Rhazali, Y. Hadi and A. Mouloudi, A Methodology of Model Transformation in MDA: from CIM to PIM, (2015) International Review on Computers and Software (IRECOS), 10(12), pp. 1186-1201.
https://doi.org/10.15866/irecos.v10i12.8088

[23] Y. Rhazali, Y. Hadi and A. Mouloudi, (2016). 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 ISBN 978-989-758- 168-7, pages 266-273.
https://doi.org/10.5220/0005690102660273

[24] Y. Rhazali, Y. Hadi, and A. Mouloudi, A Based-Rule Method to Transform CIM to PIM into MDA, International Journal of Cloud Applications and Computing, vol. 6, No. 2, 2016, pp. 11-24.
https://doi.org/10.4018/IJCAC.2016040102

[25] Y. Rhazali, Y. Hadi and A. Mouloudi, " A Model Transformation in MDA from CIM to PIM Represented by web models through SoaML and IFML," 4th International Colloquium on Information Science and Technology (IEEE CiSt'16), 2016, Tangier-Assilah, Morocco

[26] Y. Rhazali, Y. Hadi, and A. Mouloudi, Model Transformation from CIM to PIM in MDA: from Business Models defined in DFD to Design Models defined in UML, la revue des technologies de l'information, No. 9, 2016.

[27] Y. Rhazali, Y. Hadi and A. Mouloudi, " Criteria Based Evaluation for Transforming CIM to PIM," Conférence internationale JSSA'16 Journées scientifiques en science appliquées, 2016, Larache, Morocco, DOI: 10.13140/RG.2.1.3453.0165.

[28] Y. Rhazali, Y. Hadi and A. Mouloudi, " Establishing Business View and Transform it to Information System View Using DFD and UML," International conference Advanced Information Technology, Services and Systems (AIT2S-15), Settat, Morocco, DOI: 10.13140/RG.2.1.3019.6080/1.

[29] Y. Rhazali, Y. Hadi, A. Mouloudi, A Model Transformation According Model Driven Architecture, in 3rd International Conference on Computing Engineering & Enterprise Management (ICCEEM-2015), Tetouan, Morocco, 2015, pp. 6 - 14.

[30] M. Drozdova, M. Kardos, Z. Kurillova, B. Bucko, " Transformation in Model Driven Architecture," Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015, DOI 10.1007/978-3-319-28555-9_17.

[31] Mohammed Mahyoub, Anas Al-Roubaiey and Gamil Ahmed. Content-based Filter Publish Subscribe Model for Real-time WSN applications. International Journal of Ubiquitous Systems and Pervasive Networks (JUSPN). Volume 7, Issue 1 (2016), pp: 19 -27.
https://doi.org/10.5383/JUSPN.07.01.004

[32] Fatma A. Osman. Healthcare Providers' Attitudes toward Using the Technology of Smart Health. International Journal of Ubiquitous Systems and Pervasive Networks (JUSPN). Volume 6, Issue 2 (2015), pp: 11-17.
https://doi.org/10.5383/JUSPN.06.02.002