Indonesian Journal of Electrical Engineering and Computer Science
Vol. 29, No. 2, February 2023, pp. 845~851
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v29.i2.pp845-851

Transforming the business process diagram into a class diagram by model-driven architecture

Habri Mohamed Achraf, Esbai Redouane, Lamlili El Mazoui Nadori Yasser
Multidisciplinary Faculty of Nador, MASI Laboratory, Mohammed First University, Oujda, Morocco

Article Info

Article history:

Received Jun 22, 2022
Revised Sep 30, 2022
Accepted Oct 17, 2022

Keywords:

BPMN
Class diagram
Model transformation
QVT
Transformations rules

ABSTRACT

The strength of the model-driven architecture (MDA) lies in its use of models and diagrams in the process of visualizing and developing information systems of companies, that is why we chose the business process model and notations (BPMN) due to the simplicity of the symbols of its diagrams and the clarity of its notations so that all simple users of information systems within the organization can understand their needs and goals quickly and smoothly. In our paper, we suggested that we connect all categories of the staff in the companies, whatever their specialty, and the IT developers, by making a transformation between the BPMN diagrams as independent computing model to the unified modeling language (UML) class diagram as platform independent model. We chose the latter as a destination due to its development and the possibility of transforming it into a powerful and integrated program, and also to complete other transformations later, we chose query views transformations (QVT) due to its consistency with the same family of programs and previous languages.

Corresponding Author:

Habri Mohamed Achraf
Multidisciplinary Faculty of Nador, MASI Laboratory, Mohammed First University
Oujda 78, BD Berkane, Ahfir, Morocco
Email: hmohamedachraf@gmail.com

1. INTRODUCTION

In each company, there are several departments, divisions, or services that consist of human resources who have several specialties generally they are simple computer users, and they can hardly express their needs at the level of the information system. The American consortium object management group (OMG) has proposed a solution to this problem and has created a modeling language whose strong point is the ease of use, and which facilitates collaboration between the business side and the IT side by a description of the internal needs in the form of diagrams designed with simple, clear and easy to read symbols. The tool that provides graphical notations that easily describe the internal needs of the establishment and their business processes, this language based on models is business process model and notation (BPMN).

In speaking of modeling languages, the passage between models and all that is modeling we are obliged to talk about the model-driven architecture (MDA) standard created by OMG well before BPMN. This standard is based on three models which are the independent calculation model (CIM), platform independent model (PIM) and platform specific model (PSM). The performance of the MDA standard manifests itself in the protection of investments, the updating of information systems in a minimized time and allow them to migrate to adaptable platforms to new technologies, which is largely beneficial for businesses.

The idea is to use BPMN as a starting point to design the business process of the company as much as an CIM as a business process diagram and transform it into a PIM, a technically specific model that allows us to communicate with the computer side and complete the cycle of transformations later, we have chosen a

model from the same OMG family which is the class diagram of the modeling language unified modeling language (UML). We opted for the first passage of the transformation of the MDA from CIM to PIM and we intend to do the other passages thereafter, that is to say from the PIM to the PSM until being able to automate the code of the desired software. This transformation will highlight the strengths of MDA, that is to say automate the transformations starting with the purely business side of understanding its needs and also to design its information system in order to have as an optimal result which is the suitable software, among several transformation languages we chose a language created and developed by the OMG group which is the query view transformation (QVT) [1].

We rethink in our article types of transformations such as a transformation from CIM to PIM of the same modeling language, namely UML, by two different transformation tools which are QVT and ATL. Our article contains five sections, the first is the introduction the second is the related work where we talk about the papers that inspired us to do this work, and the background is presented in the third section which offers definitions of the bases used, The fourth section is the part that reveals the added value of our work, by drawing the BPMN diagram according to the case study in the first step, then by establishing the transformation rules by respecting the standards of the QVT language and finally by having a class diagram as the desired result. The last component of our work is a conclusion which is at the same time a future introduction for the tasks that we intend to accomplish in the future, which will enrich our work and even take it as a reference.

2. RELATED WORK

In this step we present the related works of the methods of transformations convergent to our idea. In [2] the authors propose two transformations the first is horizontal and called refinement CIM model between the business process diagram and the use case diagram UML, the second is vertical to the UML sequence diagram as a PIM model, using the specification that defines the rules and vocabulary: semantics business vocabulary rules (SBVR). Rhazali *et al.* [3] present a method of transformation based on the modernization of MDA, it starts by creating the transformable model at the CIM level and transforms it afterward using ATL atlas transformation language to obtain the PIM model as a result.

The authors propose in [4] a transformation between BPMN and the diagram of UML namely the activity diagram, practically there is a convergence between the business process diagram and the activity diagram which allows having precise and direct transformation rules. Macek and Richta [5] propose an idea of transformation that resembles the previous one, starting with the diagram of the business process BPMN and obtaining the activity diagram of UML but what characterizes this transformation is the use of an XML metadata exchange to represent the two models and to specify the document style as well as the vocabulary at the input and output level they used the XSLT transformation type.

Debnath *et al*. [6] have done a work that starts by designing a BPMN business process diagram that helps to create a java ee 6 business platform, in three steps starting with a transformation between the BPMN diagram and class diagram of UML, after transforming it into a diagram UML based on java and at the end get meta-object facility (MOF) script based on java ee profiles. We were also inspired by the transformation carried out from BPMN towards graphical and mathematical tools (COLORED PETRI NET) allowing us to model and verify the dynamic behavior of discrete event systems in [7] and from the Web Ontology Language obtained by a transformation of BPMN, OWL2 is a language with components that include an overview of the document, which serves as an introduction to OWL 2, describes the relationship between OWL 1 and OWL 2, and provides an entry point to deliverables remaining via a worksheet [8], after observing the two papers we find that BPMN remains the clearest and practical language.

3. BACKGROUND KNOWLEDGE

3.1. Model-driven architecture (MDA)

The MDA standard as its name suggests is a model-driven architecture that has introduced a new method of computer design, among its strengths is to protect the investment that already exists in the organization and even make it updated in a little time [9], that is to say with MDA we can work with the useful part of our information system and adapt it with new emerging technologies or even keep the software side and update the level of design and development and analysis, like all the body of tools proposed and created by OMG, the MDA approach confirms that a model is worth a thousand words [10]. The MDA models: a) independent model computation (CIM) independent calculation model, b) platform-independent model (PIM) platform-independent model. c) platform-specific model (PSM) platform-dependent model. In our work, we focus on this transition between the models provided by MDA, especially in the first place from the CIM to the PIM.

3.2. Business process model and notation: BPMN

A Model BPMN business process model and notation is a specific modeling language for the business side it offers simple, clear, and easy-to-read graphical notations. It is the ideal way to discuss an information system with a manager in a company who can express his need or his vision using diagrams drawn by basic symbols. It is considered the best notion of computer science to start with simple users, this standard was created at the beginning by the BPMI and after an arrangement with OMG and the latter has been supported and maintained BPMN since 2005 [11]–[13].

3.3. Query views transformation: QVT

As part of saving effort and reducing errors, OMG adopted the model transformation language called QVT in 2005 [14]–[16]. Model transformation into MDA is an automated way of modifying and creating models. A model transformation usually specifies the acceptable input models and the models that it can output by specifying the Meta model that a model must conform to. The QVT standard introduces the means to query, display and transform models based on MOF 2.0. The QVT language is supported by Eclipse, so on the technical side, there will be no obstacle to ensuring the passage between BPMN and UML that are already supported by Eclipse.

4. METHOD

Our approach is a solution that will highlight a relationship between simple users and IT analysts, starting by translating the need, vision, and situation of a company into a BPMN business process diagram and transforming after this initial model into a more specialized model, and more technically sophisticated, namely the UML class diagram to complete our global vision which to start by writing the model independent of calculation CIM make transformations towards the PIM and the PSM to finally have the code or the software in Figure 1, in our paper, we will carry out the first step which is to be considered as a bridge which the simple users with the computer developers in an automatic way. In our article we are inspired by other works for example the transformation called refinement at the beginning of CIM (BPMN) towards CIM (the use case diagram of UML) after their passage to carry out an another CIM to PIM transformation two diagrams of the same UML language use case diagram to UML sequence diagram, the passage between BPMN and the UML CIM to CIM activity diagram (refinement) and one can notice that there is a great resemblance between these two diagrams, in the dedescription transformation of two diagrams from UML, CIM to PIM. We chose to start with a case study represented by a BPMN diagram and to go directly to the UML class diagram which allowed us to go more on the IT development side without going through other transformations for example case diagram usage, sequence diagram, or activity diagram of UML [17]–[25].

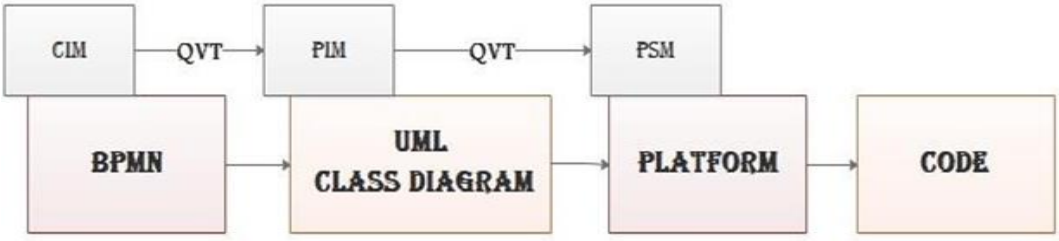


Figure 1. Our proposed transformation compared to the MDA models

4.1. The transformation rules

After reading the transformation rules between BPMN and the use case diagram, sequence diagram, or activity diagram we managed to write the rules of transformation between BPMN and the class diagram. In our case, we have a simple language BPMN and a more efficient language (UML class diagram) so we need to use conditions to direct the transformation properly. For example, the "Task" element according to the condition can be transformed into class or method, sequence flow or message flow with association according to the position of each element in the diagram and its meaning in the whole process of the case study in general shown in Table 1.

Table 1. Transformation rules for the two models

Transformation rules	Source model element	Target model element
Participant (Pool) to class	Participant (Pool)	Class
Lane to class	Lane	Class
Task “Activity” to class or method "with condition"	Task “Activity” receive task send task	Class
Sequence flow or message flow to association	Sequence Flow Message Flow	Association (condition)
Exclusive gate way (OR XOR …) to association	Exclusive gate way	Association or (Generalization)

4.2. Case study

The practical demonstration of our approach requires a case study. So, we chose online purchase which has two main participants the customer and the seller (website). The BPMN diagram below in Figure 2 contains several symbols, for example, the pool is the entity that represents buyer or seller and each pool contains the process that will do it starting from the start event, the sequence flows that show the direction of the process and that connect the elements in it, the activity or task which is action can be simple, sent or received depending on the collaboration between the two participants, for example, the "receive order" task it is an activity received by the seller after the buyer sends the "send order" task.

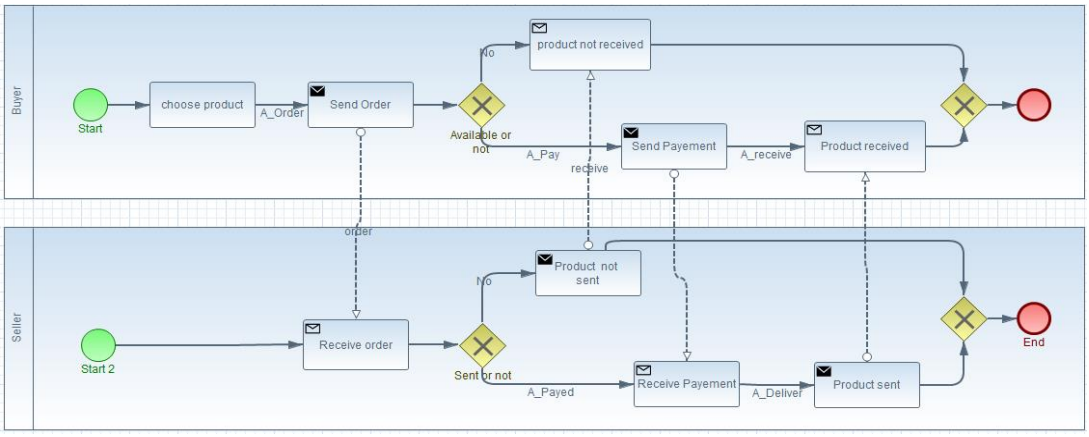
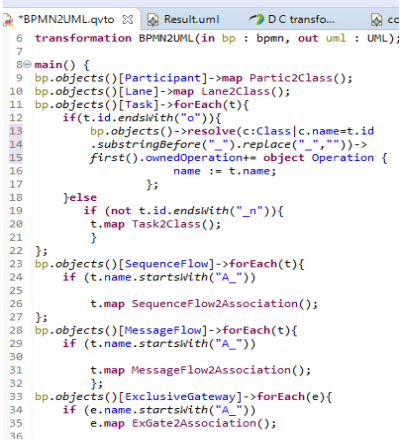


Figure 2. The process diagram of BPMN (collaboration)

4.3. The extracted QVT code

The (M2M) model to model transformation is the main step in our work. We also introduce the import of BPMN and UML sources and we begin to enter the QVT code based on the transformation rules. They are written in accordance with the standards of the QVT transformation language based on the MOF with the architecture of four layers M3, M2, M1, and M0 to ensure the passage between the two models as shown in Figure 3.



```
*BPMN2UML.qvto   Result.uml   D C transfo...   cc
transformation BPMN2UML(in bp : bpmn, out uml : UML);

main() {
bp.objects()[Participant]->map Partic2Class();
bp.objects()[Lane]->map Lane2Class();
bp.objects()[Task]->forEach(t){
    if(t.id.endsWith("o")){
        bp.objects()->resolve(c:Class|c.name=t.id
        .substringBefore("_").replace("_",""))->
        first().ownedOperation+= object Operation {
                name := t.name;
            };
    }else
        if (not t.id.endsWith("_n")){
        t.map Task2Class();
        }
};
bp.objects()[SequenceFlow]->forEach(t){
    if (t.name.startsWith("A_"))

        t.map SequenceFlow2Association();
};
bp.objects()[MessageFlow]->forEach(t){
    if (t.name.startsWith("A_"))

        t.map MessageFlow2Association();
        };
bp.objects()[ExclusiveGateway]->forEach(e){
    if (e.name.startsWith("A_"))
        e.map ExGate2Association();
```

Figure 3. A part of the QVT code

We opted for the eclipse platform which contains the plugins of all the languages used in our paper. To draw the BPMN diagram and develop the eclipse code and obtain the desired result, at the transformation level and especially between the "task" of BPMN and the class or the method of UML. In Figure 4 we used to distinguish each case according to the design by using a condition at the level of the id of the element concerned.

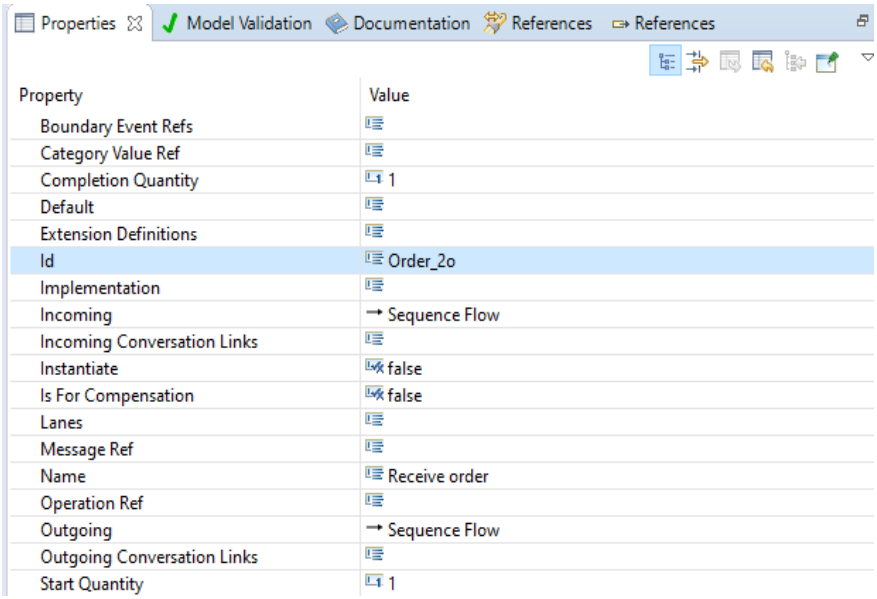


Figure 4. The properties of the "receive order" task

5. RESULTS

After executing the QVT code using eclipse. Figure 5 Shows classes, methods, and associations transformed from participants, activities, sequences flow, and messages flow. The result is obtained in the form of a UML class diagram structure composed of target elements transformed from source BPMN elements.

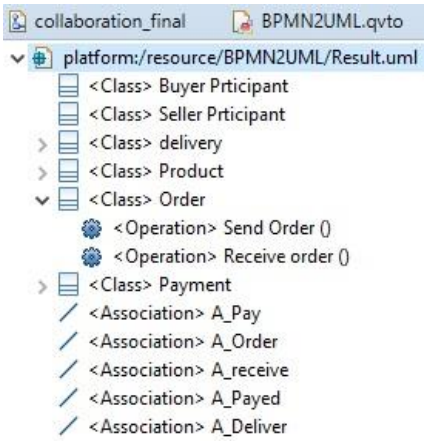


Figure 5. The class diagram structure

To reach the last stage in this work we used the "Papyrus" tool of the eclipse to draw the class diagram of UML from the previous structure. At this level we can notice all the transformations carried out for example the class "seller" is transformed from the pool "seller", the task "choose product" is transformed into a class product, the sequence flow "send_product" is transformed into association a_deliver as. Then we got a clear and well-designed class diagram automatically from the BPMN process diagram as shown in Figure 6.

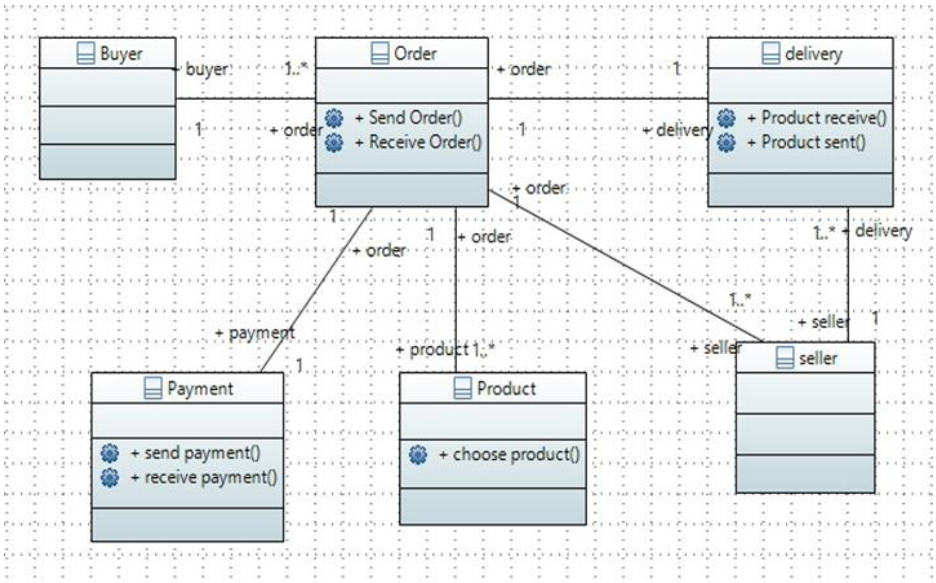


Figure 6. Our result as a class diagram

6. CONCLUSION AND PERSPECTIVES

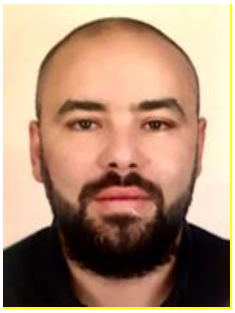
The use of MDA shows us the added value of this standard for information systems and their design, the transformations of the models provided by MDA allow having a homogeneous sequence between the components of the information system. We projected on our work by carrying out a direct and automated transformation between the "collaboration" process diagram of BPMN and the class diagram of UML assuming that we create a gateway between the simple users on the business side with the developers in the computer engineering to achieve the company's objective, which is investment protection, saving time and, above all, having a solid and up-to-date information system. We can develop our result by going to the step of transforming the class diagram into another model specific to the PSM platform, or make transformations in the opposite direction within the framework of the ADM, for example, one can change a tool already used to be able to make the comparison.

REFERENCES

[1] X. Blanc, "l'architecture MDA," *MDA en action Ingénierie logicielle guidée par les modèles*. 2005.
[2] N. Addamssiri, A. Kriouile, Y. Balouki, and G. Taoufiq, "Generating the PIM Behavioral Model from the CIM using QVT," *J. Comput. Sci. Inf. Technol.*, vol. 2, no. 3 & 4, 2014, doi: 10.15640/jcsit.v2n3-4a4.
[3] Y. Rhazali, Y. Hadi, and A. Mouloudi, "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*, 2016, pp. 266–273. doi: 10.5220/0005690102660273.
[4] N. Bảo, "A proposal for a method to translate BPMN model into UML activity diagram," 2010. [Online]. Available: http://www.nqbao.com/archives/files/BPMN-UMLAD.pdf
[5] O. Macek and K. Richta, "The BPM to UML activity diagram transformation using XSLT," in *CEUR Workshop Proceedings*, 2009, vol. 471, pp. 119–129.
[6] N. Debnath, C. A. Martinez, F. Zorzan, D. Riesco, and G. Montejano, "Transformation of business process models BPMN 2.0 into components of the Java business platform," in *IEEE 10th International Conference on Industrial Informatics*, Jul. 2012, pp. 1035–1040. doi: 10.1109/INDIN.2012.6300914.
[7] C. Dechsupa, W. Vatanawood, and A. Thongtak, "Transformation of the BPMN design model into a colored petri net using the partitioning approach," *IEEE Access*, vol. 6, pp. 38421–38436, 2018, doi: 10.1109/ACCESS.2018.2853669.
[8] M. Kchaou, W. Khlif, F. Gargouri, and M. Mahfoudh, "Transformation of BPMN model into an OWL2 ontology," in *Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering*, 2021, pp. 380–388. doi: 10.5220/0010479603800388.
[9] O. Betari, S. Filali, A. Azzaoui, and M. A. Boubnad, "Applying a model driven architecture approach: transforming CIM to PIM using UML," *Int. J. Online Eng.*, vol. 14, no. 09, p. 170, Sep. 2018, doi: 10.3991/ijoe.v14i09.9137.
[10] R. Ambara and A. N. Fajar, "Business process management for system development," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 2, pp. 1331–1334, 2019.
[11] R. Braun and W. Esswein, "Classification of domain-specific bpmn extensions," in *Lecture Notes in Business Information Processing*, 2014, vol. 197, pp. 42–57. doi: 10.1007/978-3-662-45501-2_4.
[12] M. Argañaraz, A. Funes, and A. Dasso, "An MDA approach to business process model transformations," *SADIO Electron. J. Informatics Oper. Res.*, vol. 9, no. 1, pp. 24–48, 2010.
[13] S. Gotti and S. Mbarki, "IFVM bridge: a model driven IFML execution," *Int. J. Online Biomed. Eng.*, vol. 15, no. 04, p. 111, Feb. 2019, doi: 10.3991/ijoe.v15i04.9707.

[14] I. Arrassen, R. Esbai, A. Meziane, and M. Erramdani, "QVT transformation by modeling: from UML model to MD model," in *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Mar. 2012, pp. 86–91. doi: 10.1109/SETIT.2012.6481895.
[15] R. Esbai, F. Elotmani, and F. Z. Belkadi, "Toward automatic generation of column-oriented NoSQL databases in big data context," *Int. J. Online Biomed. Eng.*, vol. 15, no. 09, p. 4, Jun. 2019, doi: 10.3991/ijoe.v15i09.10433.
[16] M. A. Habri, R. Esbai, and Y. L. E. M. Nadori, "BPMN to UML class diagram using QVT," 2022, pp. 593–602. doi: 10.1007/978-981-16-3637-0_42.
[17] J. Irazábal, C. Pons, and C. Neil, "Model transformation as a mechanism for the implementation of domain specific transformation languages," *SADIO Electron. J. Informatics Oper. Res.*, vol. 9, no. 1, pp. 49–66, 2010.
[18] F. Abdelhedi, A. Ait Brahim, F. Atigui, and G. Zurfluh, "Processus de transformation MDA d'un schéma conceptuel de données en un schéma logique NoSQL," *INFORSID 2016 - Actes du 8e Forum Jeunes Cherch. du Congr. INFORSID*, pp. 15–30, 2016.
[19] Specification, "UML Infrastructure Final Adopted Specification," 2004.
[20] R. Kraleva, V. Kralev, N. Sinyagina, P. Koprinkova-Hristova, and N. Bocheva, "Design and analysis of a relational database for behavioral experiments data processing," *Int. J. Online Eng.*, vol. 14, no. 2, pp. 117–132, 2018, doi: 10.3991/ijoe.v14i02.7988.
[21] Y. Rhazali, Y. Hadi, and A. Mouloudi, "Model Transformation with ATL into MDA from CIM to PIM structured through MVC," *Procedia Comput. Sci.*, vol. 83, pp. 1096–1101, 2016, doi: 10.1016/j.procs.2016.04.229.
[22] J. Steel and J.-M. Jézéquel, "Typing relationships in MDA," in *In Proc. of the 2th Second European Workshop on Model Driven Architecture (MDA)*, 2004, pp. 154–159.
[23] A. Kriouile, "An MDA Method for Automatic Transformation of Models from CIM to PIM," *Am. J. Softw. Eng. Appl.*, vol. 4, no. 1, p. 1, 2015, doi: 10.11648/j.ajsea.20150401.11.
[24] "Transformation de modèles et Ingénierie Dirigées par les Modèles (IDM ou MDE Model Driven Engineering ou bien encore MDA Model Driven Architecture)," *urbanisation-si.com*. https://www.urbanisation-si.com (accessed Apr. 07, 2022).
[25] Y. Rhazali, Y. Hadi, and A. Mouloudi, "CIM to PIM transformation in MDA: from service-oriented business models to web-based design models," *Int. J. Softw. Eng. Its Appl.*, vol. 10, no. 4, pp. 125–142, Apr. 2016, doi: 10.14257/ijseia.2016.10.4.13.

BIOGRAPHIES OF AUTHORS



Habri Mohamed Achraf is Associate Ph.D. Student, he got his specialized master's degree in software quality from Abdel Malek Essaadi University at tangier. He is a researcher studying the business process model and notation and their applications at MASI laboratory in Mohammed First University, Morocco. He can be contacted at email: hmohamedachraf@gmail.com.

Prof. Esbai Redouane received teaches the concept of Information systems at Mohammed 1 University. He got the thesis of his national doctorate in 2012. He got a degree as an engineer in Computer Sciences from the National School of Applied Sciences at Oujda. He received his M.Sc. degree in New Information and Communication Technologies from the faculty of sciences and Techniques at Sidi Mohamed Ben Abdellah University. His activities of research in the MASI Laboratory (Applied Mathematics and Information System) focused on MDA (Model Driven Architecture) integrating new technologies XML, Spring, Struts, and GWT. He can be contacted at email: es.redouane@gmail.com.

Lamlili El Mazoui Nadori Yasser his is a Dr. at Mohammed First University in the Faculty of Sciences. He got a degree in engineering in Computer Sciences from the National School of Applied Sciences at Oujda. He received his M.Sc. degree in New Information and Communication Technologies from the faculty of sciences and Techniques at Sidi Mohamed Ben Abdellah University. His research activities at the MATSI Laboratory (Applied Mathematics, Signal Processing, and Computer Science) have focused on web marketing in social networks using the MDA (Model Driven Architecture) approach. He can be contacted at email: lamliliyasser@gmail.com.