Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 1, October 2025, pp. 216~224
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i1.pp216-224

Automating electronic document management design: a model-driven approach using business process

Soufiane Hakkou, Redouane Esbai, Yasser Lamlili El Mazoui Nadori
Laboratory of Applied Mathematics and Information Systems, Mohammed First University, Oujda, Morocco

Article Info

Article history:

Received Dec 5, 2024
Revised Apr 30, 2025
Accepted Jul 3, 2025

Keywords:

Automating EDM
BPMN diagrams
Business process
Document workflows
Electronic document system
Transformation model

ABSTRACT

Model-driven architecture (MDA) is a useful approach for designing enterprise information systems through structured models. This study applies MDA to electronic document management (EDM) systems, which are essential for improving document workflows and ensuring regulatory compliance. Organizations often face difficulties when converting business process models into software-ready designs. Current transformation methods are complex, involving multiple intermediate steps that increase effort and risk of errors. The objective of this work is to create a direct transformation from business process model and notation (BPMN) diagrams to unified modeling language (UML) class diagrams. This aims to improve automation, reduce modeling effort, and maintain consistency. The proposed methodology uses MDA principles and query/view/transformation (QVT) to automatically map BPMN elements to UML classes based on predefined rules. The approach is implemented within the eclipse modeling framework (EMF) and validated through a case study on EDM systems. The transformation successfully generates UML class diagrams that accurately represent BPMN-based business processes. The results demonstrate: increased automation, reducing manual effort in software modeling, improved model consistency, eliminating errors associated with multi-step transformations and enhanced business-IT alignment, providing a structured approach for business professionals and developers.

Corresponding Author:

Soufiane Hakkou
Laboratory of Applied Mathematics and Information Systems, Mohammed First University
Mohammed V Avenue, P.O. Box 658, Oujda 60000, Morocco
Email: soufianehakkou@yahoo.fr

1. INTRODUCTION

Organizations across various sectors increasingly rely on electronic document management (EDM) systems to streamline workflows, ensure regulatory compliance, and enhance document accessibility. However, bridging the gap between business requirements and IT implementations remains a challenge. Business process model and notation (BPMN) has emerged as a widely adopted graphical language to visually represent business processes, yet its integration with software development methodologies requires further exploration. Despite the widespread use of BPMN for modeling business processes, transforming these models into software-ready representations remains a challenge. Current approaches often involve multiple intermediate steps, increasing complexity and reducing automation potential. Our study aims to address this challenge by proposing a direct transformation from BPMN diagrams computation-independent models (CIM) into unified modeling language (UML) class diagrams platform-independent models (PIM), following the model-driven architecture (MDA) framework [1].

Several studies have explored BPMN-to-UML transformations to improve software development workflows:

- Addamssiri *et al.* [2] proposed a two-step transformation methodology involving UML use case diagrams and sequence diagrams to refine business process representations.
- Rhazali *et al.* [3] developed an MDA-based approach using the atlas transformation language (ATL) to transition from CIM to PIM, improving precision and adaptability.
- Bảo examined BPMN-to-UML activity diagram transformations, highlighting their conceptual alignment for process modeling [4], [5].
- Debnath *et al.* [6] demonstrated a multi-step BPMN transformation approach leading to Java EE platform implementation, showcasing the potential for automated code generation [7].
- Kchaou *et al.* [8] expanded BPMN applications by transforming them into OWL2 ontologies for semantic reasoning and knowledge representation [9].

While these contributions have advanced model transformation techniques, none provide a direct, automated BPMN-to-UML class diagram transformation using query/view/transformation (QVT), which remains a critical gap in the literature. Despite prior efforts, existing methodologies introduce complexity through intermediate transformations (e.g., activity diagrams, sequence diagrams) before reaching UML class diagrams. This study aims to:

- Automate the transformation from BPMN to UML class diagrams, eliminating unnecessary intermediary steps.
- Ensure model consistency through well-defined transformation rules aligning with object management group (OMG) standards.
- Enhance collaboration between business analysts and developers by providing a direct mapping of business concepts to software structures.

In this study, we focus on transforming BPMN diagrams, which represent CIMs, into UML class diagrams, a type of PIM. This transformation bridges the business and technical domains, allowing for subsequent transitions to PSMs and automated software generation. Using QVT, a language designed by OMG, we aim to demonstrate a streamlined, automated process that facilitates collaboration and enhances the overall design of information systems, with EDM systems serving as a practical case study.

To address these gaps, we propose a model-driven transformation process using QVT within the eclipse modeling framework (EMF). The following sections outline our methodology:

- Section 2 provides background knowledge on BPMN, UML, and QVT.
- Section 3 describes the proposed transformation methodology and its implementation.
- Section 4 presents a case study demonstrating the approach.
- Section 5 discusses the results, highlighting automation benefits and potential limitations.
- Section 6 concludes with insights on future enhancements and applications.

By integrating BPMN directly into UML class diagrams, our approach simplifies model transformations, enhances automation, and improves business-IT alignment, ultimately contributing to the broader field of model-driven software engineering.

2. BACKGROUND KNOWLEDGE

2.1. Business process model and notation (BPMN)

BPMN is a graphical language developed for modeling business processes. Known for its clarity and simplicity, it provides a set of standardized symbols that enable non-technical users to visualize and communicate business needs effectively [10]. Initially created by the business process management initiative (BPMI), BPMN is now maintained by OMG. It serves as a bridge between business professionals and IT developers, facilitating a shared understanding of organizational processes [11]-[13]. This study leverages BPMN as the starting point for capturing business processes in a CIM, making it an ideal foundation for transitioning into more technically detailed models like UML class diagrams.

2.2. Query/view/transformation

QVT is an OMG-standard language designed for model-to-model transformations within the MDA framework. Introduced in 2005 [14]-[18], QVT ensures automated, consistent transformations between models. Key features of QVT include:

- Automation: reduces human error and increases efficiency during model transformation.
- Meta-model compliance: ensures input and output models adhere to predefined meta-models.
- Integration: supported by platforms like eclipse, allowing seamless transformations between models like BPMN and UML [19]-[23].

By using QVT in this study, we implement a reliable process for converting BPMN diagrams into UML class diagrams, demonstrating its utility in bridging the business and technical domains.

3. PROPOSED METHOD

Our approach aims to establish an automated transformation process that bridges the gap between non-technical users and IT developers by converting BPMN diagrams into unified UML class diagrams. This transformation follows the MDA principles, transitioning from a CIM to a PIM.

3.1. Overview of the proposed methodology

The proposed methodology focuses on using BPMN diagrams to represent business processes and then transforming them into UML class diagrams. The transformation process automates the transition from CIM to PIM, enabling clear communication between business professionals and developers. Figure 1 illustrates the transformation sequence, which bypasses intermediate models (e.g., use case, sequence, or activity diagrams) to directly obtain UML class diagrams, saving time and enhancing accuracy [24]-[27].

The chosen methodology follows the MDA framework to automate the transformation of business process models into software-ready structures. The key justifications for this approach include:

- Alignment with OMG standards: the use of BPMN and UML ensures compatibility with widely accepted modeling standards.
- Automation and consistency: QVT enables automated, rule-based transformations, minimizing human error and improving model consistency.
- Reduction of intermediary steps: unlike previous studies that require multiple intermediate transformations (e.g., UML activity diagrams), our approach directly maps BPMN elements to UML class structures.
- Practical application: the methodology is tested using real-world EDM processes, demonstrating its effectiveness in a business environment.

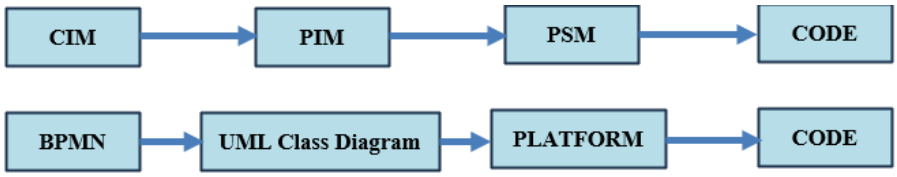

Figure 1. Our proposed transformation compared to the MDA models

3.2. Transformation rules

The transformation rules define how elements in BPMN diagrams correspond to elements in UML class diagrams. Table 1 outlines these rules, specifying the conditions for mapping BPMN components (e.g., tasks, pools, and sequence flows) to their UML equivalents (e.g., classes, methods, and associations). For instance:

- Pools in BPMN are transformed into UML classes.
- Lanes become subclasses to represent organizational hierarchies.
- Tasks may translate into methods or classes based on their role and position.
- Sequence and message flows are converted into UML associations or generalizations, depending on the relationships they depict.

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)

3.3. Case study

To demonstrate our approach, we used an EDM process as a case study. The BPMN diagram (Figure 2) illustrates the interactions between two primary participants: the document requester and the document manager. Key elements include:

- Pools representing participants (e.g., user, machine).
- Tasks representing activities (e.g., "scan file," "receive file", "evaluate", and "result of scan").
- Sequence flows indicating the progression of tasks through the document management workflow.

The transformation applies the defined rules to convert this BPMN diagram into a UML class diagram, capturing the structural and behavioral aspects of the EDM system for further development. Figure 2 illustrates the BPMN process diagram (collaboration), detailing the interactions between document requesters and document managers. This diagram provides an overview of workflow transitions within the EDM system.

Experimental setup:

- The transformation of BPMN diagrams into UML class diagrams follows the MDA approach. The setup includes the following key components.

Software and tools:

- EMF: used for model definition and transformation.
- Papyrus for UML: a modeling tool to visualize UML class diagrams.
- Eclipse QVTo: an implementation of the QVT language used for model-to-model transformation.
- BPMN 2.0 editor: for designing and validating BPMN diagrams before transformation.

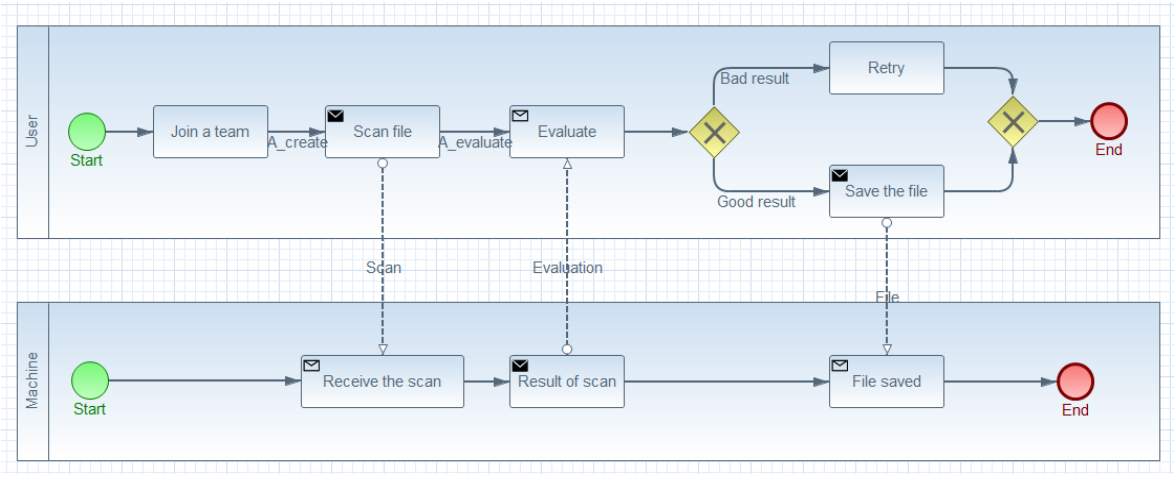

Figure 2. The process diagram of BPMN (collaboration)

3.4. Implementation

The transformation process employs the QVT language within the eclipse platform.

- Step 1: import BPMN and UML meta-models into eclipse.
- Step 2: develop QVT code following the transformation rules.
- Step 3: execute the QVT script to generate a UML class diagram from the BPMN diagram.

Figure 3 illustrates a portion of the QVT code, showcasing how BPMN elements are mapped to UML equivalents. The eclipse papyrus plugin is used to visualize the resulting UML class diagram.

4. RESULTS AND DISCUSSION

The output of the transformation is a well-structured UML class diagram containing:

- Classes and methods derived from BPMN pools and tasks.
- Associations derived from sequence and message flows.

This automated process ensures consistency, reduces manual errors, and serves as a foundation for further transformations, such as generating platform-specific models (PSMs) or code.

Figure 4 showcases the UML class diagram structure derived from the BPMN transformation process. It highlights the relationships and associations between different components within the EDM system.

```
modeltype UML uses 'http://www.eclipse.org/uml2/5.0.0/UML';
modeltype bpmn uses 'http://www.omg.org/spec/BPMN/20100524/MODEL';

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

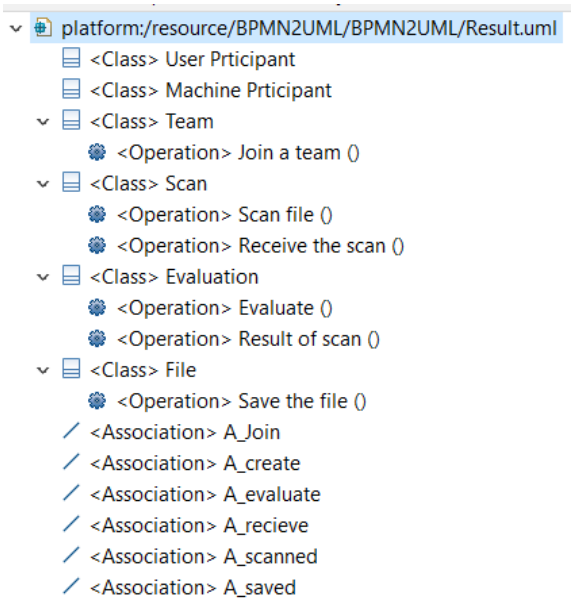

Figure 4. The class diagram structure

The transformation process successfully demonstrated the practical application of the MDA framework by automating the transition from BPMN diagrams to UML class diagrams. This transformation addressed a key challenge in aligning business requirements with technical implementation, providing a streamlined bridge between non-technical stakeholders and developers. The resulting UML class diagram effectively captured the structural and behavioral elements of the BPMN diagram, such as:

- Classes derived from participants and lanes, representing key organizational roles.
- Methods generated from tasks, detailing specific actions or processes.
- Associations formed from sequence and message flows, reflecting interactions and dependencies between components.

The use of QVT proved instrumental in ensuring accuracy and consistency during the transformation process. By adhering to predefined rules and meta-models, the approach minimized manual intervention, reducing the potential for errors. Furthermore, the case study on an online purchase process illustrated the flexibility of the methodology in handling real-world business scenarios.

The transformation outcome highlights several benefits:

1. Improved collaboration: by converting BPMN diagrams into a more technical UML class diagram, the approach fosters better communication between business professionals and IT developers.
2. Automation: the use of QVT and eclipse eliminates redundant manual work, saving time and effort.
3. Scalability: the resulting UML class diagram provides a foundation for further transformations, including PSMs and automated code generation.

However, some limitations were observed. The transformation process, while robust, relies on the accuracy of the input BPMN diagrams. Any ambiguities or inconsistencies in the initial model can propagate to the UML output. Future enhancements could address this by integrating error-checking mechanisms or additional transformation rules for more complex BPMN constructs.

Overall, the results validate the feasibility and utility of the proposed methodology, demonstrating its potential to enhance model-driven development workflows across various domains.

- The resulting UML class diagrams were evaluated for consistency with the original BPMN model.
- The validity of the transformation was assessed using Papyrus for UML.
- The correctness of class structures and relationships was manually verified by domain experts.

Figure 5 displays the final UML class diagram, representing the complete transformation from BPMN. This diagram serves as the primary outcome of the proposed methodology, illustrating how business processes are translated into structured software models.

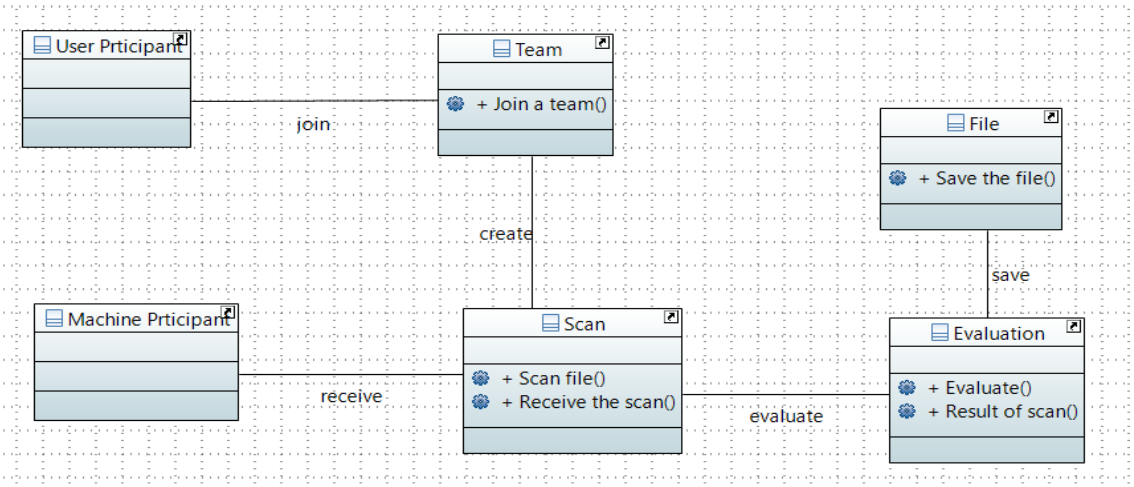

Figure 5. Our result as a class diagram

By following this structured methodology, we provide a rigorous, automated, and validated process for transforming BPMN business process models into UML class diagrams, supporting model-driven software development and business-IT alignment

In comparison to prior methodologies that involve intermediate transformations (e.g., use case and sequence diagrams), our direct BPMN-to-UML class transformation eliminates unnecessary steps, improving efficiency and reducing complexity. Studies such as Addamssiri *et al.* [4] and Rhazali *et al.* [4]

employed multi-step transformations, increasing the risk of model inconsistencies. Our approach ensures direct mapping using well-defined QVT transformation rules, enhancing model integrity. While the proposed methodology demonstrates effectiveness, several areas warrant further exploration:

- Extending the transformation process: future work could focus on extending the methodology to generate PSMs and executable code.
- Enhancing error handling: incorporating automated validation mechanisms to detect and correct inconsistencies in BPMN models before transformation.
- Applying machine learning techniques: exploring AI-driven optimization of transformation rules to improve accuracy and adaptability across different business domains.

5. CONCLUSION

This study highlights the value of MDA in designing and transforming enterprise information systems, emphasizing the transition from business process modeling to technical implementation. By automating the transformation from BPMN diagrams to UML class diagrams, the proposed approach bridges the gap between non-technical users and IT developers. The transformation not only ensures a seamless flow of information but also contributes to investment protection, time savings, and the creation of robust, up-to-date information systems.

The use of QVT has proven effective in achieving accurate and consistent model transformations while reducing the effort required for manual intervention. The results demonstrate the practical benefits of this method, including improved collaboration, automation, and scalability, as evident in the case study of an online purchase process. This research contributes to the advancement of model-driven engineering (MDE) by providing a streamlined, automated methodology for BPMN-to-UML class transformation. The approach is particularly beneficial for organizations looking to enhance software modeling efficiency while maintaining business process fidelity.

Future research will focus on extending this methodology to achieve further transformations within the MDA framework. These include transitioning from UML class diagrams to PSMs and ultimately generating executable code. Additionally, the reverse transformation process, as part of the architecture-driven modernization (ADM) framework, could be explored to compare and evaluate various tools and approaches. Such advancements would enhance the adaptability and flexibility of the proposed solution, further enriching its applications in model-driven software engineering.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Soufiane Hakkou	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Redouane Esbai	✓	✓	✓	✓	✓	✓	✓	✓					✓	
Yasser Lamlili El Mazoui Nadori	✓		✓	✓			✓			✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] H. M. Achraf, E. Redouane, and L. El Mazoui Nadori Yasser, "Transforming the business process diagram into a class diagram by model-driven architecture," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 2, pp. 845–851, Feb. 2023, doi: 10.11591/ijeecs.v29.i2.pp845-851.

[2] N. Addamssiri, A. Kriouile, Y. Balouki, and G. Taoufiq, "Generating the PIM behavioral model from the CIM using QVT," *Journal of Computer Science and Information Technology*, 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," Vietnamese German University, 2010.

[5] O. Macek and K. Richta, "The BPM to UML activity diagram transformation using XSLT," *CEUR Workshop Proceedings*, vol. 471, pp. 119–129, 2009.

[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 International Conference on Industrial Informatics (INDIN)*, 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 *International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE - Proceedings*, 2021, vol. 2021-April, 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," *International Journal of Online Engineering*, vol. 14, no. 9, pp. 170–181, Sep. 2018, doi: 10.3991/ijoe.v14i09.9137.

[10] R. Ambara and A. N. Fajar, "Enterprise service bus (ESB) and business process management for system development," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 2, pp. 1331–1334, Dec. 2019, doi: 10.35940/ijeat.A9850.129219.

[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] LNCS FAQ," BPModeling.com. [Online]. Available: http://www.bpmodeling.com/faq/lncs. [Accessed: Nov. 2, 2021].

[13] M. Argañaraz, A. Funes, and A. Dasso, "An MDA approach to business process model transformations," *SADIO Electronic Journal of Informatics and Operations Research*, vol. 9, pp. 24–48, Sept. 2010. [Online]. Available: https://revistas.unlp.edu.ar/EJIO/article/view/17546.

[14] S. Gotti and S. Mbarki, "IFVM bridge: a model driven IFML execution," *International journal of online and biomedical engineering*, vol. 15, no. 4, pp. 111–126, Feb. 2019, doi: 10.3991/ijoe.v15i04.9707.

[15] I. Arrassen, A. M. -, R. S. -, and M. E. -, "QVT transformation by modelling - from UML model to MD model," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 5, 2011, doi: 10.14569/ijacsa.2011.020502.

[16] R. Esbai, F. Elotmani, and F. Z. Belkadi, "Toward automatic generation of column-oriented NoSQL databases in big data context," *International journal of online and biomedical engineering*, vol. 15, no. 9, pp. 4–16, Jun. 2019, doi: 10.3991/ijoe.v15i09.10433.

[17] M. A. Habri, R. Esbai, and Y. L. El Mazoui Nadori, "BPMN to UML class diagram using QVT," in *Smart Innovation, Systems and Technologies*, vol. 237, 2022, pp. 593–602.

[18] blanc Xavier, "MDA en action - ingénierie logicielle guide par les modèles," *Eyrolles*, 2005.

[19] J. Irazábal, C. Pons, and C. Neil, "Model transformation as a mechanism for the implementation of domain specific transformation languages," *SADIO electronic journal of informatics operations research*, vol. 9, no. 1, pp. 49–66, 2010.

[20] 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 Chercheurs du Congres INFORSID*, pp. 15–30, 2016.

[21] Object Management Group, "UML Infrastructure, v2.0, Final Adopted Specification," Sept. 2003. [Online]. Available: https://www.omg.org/spec/UML/2.0/Infrastructure/PDF

[22] R. Kraleva, V. Kralev, N. Sinyagina, P. Koprinkova-Hristova, and N. Bocheva, "Design and analysis of a relational database for behavioral experiments data processing," *International Journal of Online Engineering*, vol. 14, no. 2, pp. 117–132, Feb. 2018, doi: 10.3991/ijoe.v14i02.7988.

[23] Y. Rhazali, Y. Hadi, and A. Mouloudi, "Model transformation with ATL into MDA from CIM to PIM structured through MVC," *Procedia Computer Science*, vol. 83, pp. 1096–1101, 2016, doi: 10.1016/j.procs.2016.04.229.

[24] Steel and J.-M. Jézéquel, "Typing relationships in MDA," in *Proc. 2nd European Workshop on Model Driven Architecture (MDA) with an Emphasis on Methodologies and Transformations*, no. 17–4, pp. 154–159, 2004. [Online]. Available: http://www.cs.kent.ac.uk/pubs/2004/2028.

[25] A. Kriouile, "An MDA method for automatic transformation of models from CIM to PIM," *American Journal of Software Engineering and Applications*, vol. 4, no. 1, p. 1, 2015, doi: 10.11648/j.ajsea.20150401.11.

[26] Y. Rhazali, Y. Hadi, and A. Mouloudi, "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, Apr. 2016, doi: 10.14257/ijseia.2016.10.4.13.

[27] R. Esbai, S. Hakkou, and M. A. Habri, "Modeling and automatic generation of data warehouse using model-driven transformation in business intelligence process," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 3, pp. 1866–1874, Jun. 2023, doi: 10.11591/ijeecs.v30.i3.pp1866-1874.

BIOGRAPHIES OF AUTHORS

Soufiane Hakkou is associate Ph.D. student, he got his specialized master's degree in software quality from Mohammed First University at Oujda. He is a researcher studying the business process model and notation and their applications in Electronic Document Management using MDA at MASI laboratory in Mohammed First University, Morocco. He can be contacted at email: soufianehakkou@yahoo.fr.

Redouane Esbai teaches the concept of Information System at Mohammed First University. He got his thesis for his national doctorate in 2012. He got a degree of an engineer in Computer Sciences from the National School of Applied Sciences at Oujda. His research activities in the MASI Laboratory (Applied Mathematics and Information System) focusing on MDA integrating new technologies XML, Spring, Struts, and GWT. He can be contacted at email: es.redouane@gmail.com.

Yasser Lamlili El Mazoui Nadori he 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 approach. He can be contacted at email: lamliliyasser@gmail.com.