Bulletin of Electrical Engineering and Informatics
Vol. 11, No. 5, October 2022, pp. 2773~2782
ISSN: 2302-9285, DOI: 10.11591/eei.v11i5.3973

A methodology for transforming BPMN to IFML into MDA

Abir Sajji[1], Yassine Rhazali[2], Youssef Hadi[1]
[1]Computer Research Laboratory, Faculty of Science, Ibn Tofail University, Kenitra, Morocco
[2]Information and Communication Systems Engineering Research Group, Higher School of Technology, Moulay Ismail University, Meknes, Morocco

Article Info

Article history:

Received Apr 18, 2022
Revised Jun 10, 2022
Accepted Jul 15, 2022

Keywords:

BPMN
CIM
GUI
IFML
MDA
Model transformations
PIM

ABSTRACT

Responding to rising information system complexity and the high expense of technological migration, model driven architecture (MDA) was created. As a result, the OMG advocates raising the abstraction level to overcome technological limitations. MDA seeks to describe the functional and performance requirements of an application on a platform independently. Using the MDA approach, the business process model and notation (BPMN), and interaction flow modeling language (IFML) standards, we represent a methodology that allows transforming semi-automatically from the computation independent model (CIM) level to the platform independent model (PIM) level; to achieve this a collection of unique rules for transforming in a semi-automatic manner from CIM to PIM were developed. At the CIM level, we create models of business process using the notation standard BPMN, and IFML is used to adapt PIM models with web-oriented graphical user interfaces (GUI). To properly demonstrate the transformation procedure from CIM to PIM models a case study of the order management process was presented.

Corresponding Author:

Abir Sajji
Computer Research Laboratory, Faculty of Science, Ibn Tofail University in Kenitra
B.P 242 Kenitra, Morocco
Email: abir.sajji@uit.ac.ma

1. INTRODUCTION

The establishment of connections between distinct models is essential in model driven architecture (MDA) [1], and this is automatically created via model transformations. The main model transformations in MDA are the transformation from computation independent model (CIM) to platform independent model (PIM) levels and from PIM to PSM levels. The initial model transformation provided by MDA is CIM to PIM, which allows a PIM model to be created from a CIM model by including technical data in the CIM. Then, the PIM to PSM model transformations allows the inclusion of technical information related to an objective platform to PIM. Transforming in a semi-automatic manner from PIM level to PSM level is allowed by some standards. However, no standard currently supports the CIM to PIM transformation. There are many similarities between these two levels which is why numerous searches describe many approaches concerning PIM to PSM model transformations, for the reason that these models are so dissimilar, CIM to PIM transformation is infrequently covered in academic topics.

We seek to elevate CIM models beyond the level of a simple document designed by business leaders and implemented by software experts. Instead, we want to create several productive models for CIM as a result, we will be able to provide PIM models semi-automatically. In this article, a group of selected rules for a semi-automatic transformation from the CIM model to the PIM model is developed. We utilize the business process model and notation (BPMN) to construct business process models at the CIM level that BPMN is an

OMG supported standard for business process modeling. Following that, for graphical user modeling, we utilize IFML, an OMG standard, to transform CIM to PIM in a semi-automated approach by combining PIM models with web-based user interfaces. Our methodology is based on rules transformation and expressed in atlas transformation language (ATL).

Our approach outlines unique transformation rules. The purpose of this search is to develop a rich PIM level. Our approach was applied to a case study of the order management process up to final delivery to validate our ideas and the IFML model is created by the use of transformation rules starting from the BPMN model. The paper is structured as; the relevant works relating to CIM to PIM transformation are presented in section 2. The rules transformation that enables the transition from CIM to PIM are described in section 3 of our proposal. In section 4, we show an example of our methodology in a case study. Section 5 shows discussion and limitations; finally, in section 6, we conclude and specify the upcoming work.

2. RELATED WORK

This section explores the related work about CIM to PIM transformation models on the MDA approach. AutoCRUD is a plug-in for WebRatio that automates the creation of CRUD operations in IFML, according to R-Echeverria *et al*. [2]. This tool increases Web developer efficiency by generating IFML specifications for CRUD operations in WebRatio automatically.

The purpose of the research of Huang *et al*. [3] is about GUI models using static reverse engineering from Android source code. The authors intended to generate GUI models statically for Android applications. The advantages of this approach are static analysis, as compared to dynamic methods, may effectively extract models to better reflect the implementation of application code, which can help many professions such as model-based testing and code generation.

Waheed *et al*. [4] describe a method for stress testing web applications that involve automated script generation. The suggested solution comprises a meta-model that can be automated, as well as model-based construction of a system/tool that can generate test scripts for stress testing web applications using IFML. This solution will save users time by eliminating the need for manual script production and also assisting developers in their overall development.

R-Echeverria *et al*. [5] represent a method for IFML, to eventually automate repetitious specification tasks. This approach is defined by implementing IFML patterns for CRUD operations, the advantages of this approach are a significant reduction in the amount of time spent on repetitive tasks also errors have been reduced significantly as a result of redundant IFML specifications, and finally a rising consistency in CRUD modeling across projects making applications more regular and useable.

Hamdani *et al*. [6] define a novel approach for generating IFML models from the text-based specifications automatically using natural language processing (NLP) characteristics, a collection of NLP rules is established to extract critical IFML elements such as view components, events, and so on from textual requirements, and case studies of a movie manager and an online bookstore are used to examine the efficacy of the proposed framework.

Laaz *et al*. [7] describe a new modeling technique based on a model-driven development process that begins by representing BPMN and ODM models as CIM Models and after transforms them into an IFML diagram using a transformation engine, from which final code may be easily generated.

The overall review of previous research showed that all of those studies discussed MDA's CIM to PIM transformation using IFML standard, but the semi-automatic generation of graphical interfaces is never discussed before, when compared to typical software development methodologies, our research treats a new approach in MDA that uses BPMN and IFML standards and which will enable the transformation from BPMN to IFML by applying the developed transformation rules.

3. CIM TO PIM TRANSFORMATION APPROACH

The first step in application development is to develop a customer’s specifications. The CIM model is unaffected by the technological details of the application and includes business process specifications as well as the customer's requirements [1], conforming to OMG, BPMN [8] is a specific standard for designing business processes. We used BPMN to represent business processes at the CIM level in our approach.

BPMN is the business process modeling standard, and all of the advantages of business process modeling standards are focused on BPMN. Then, we transform CIM models into PIM models through web-oriented user interfaces using IFML [9], [10]; an OMG standard for user interface modeling; is a graphical modeling language used to define front-end application content, interface composition, user interaction, and control behavior for a range of applications such as web, mobile, and desktop software. Our methodology protects business knowledge throughout the PIM transformation, allowing the development of a high-quality information system.

A set of transformation rules underpin the model transformations. Human language is used to describe these rules that are represented graphically to illustrate the transformation process. After that, the IFML model is produced as a consequence of semi-automatic CIM model transformations, the transformation is done using unique rules that are also expressed in ATL language.

3.1. CIM to PIM transformation rules

Table 1 (see in appendix) shows the rules for transforming BPMN models into IFML models. Every transforming rule is represented by a schema and is specified in both ATL and human language. To express transformation rules in ATL, the CIM level metamodel and the PIM level metamodel must first be defined.

4. CASE STUDY

This study describes the order management process up to final delivery to present the shifting methodology from the CIM level to the PIM level. This case study's objective is to manage customer orders. The process begins with the reception of the customer’s order, then the salesperson can check the stock situation, if the product is not in stock, the salesperson contacts the supplier to order the product, and the latter can quickly communicate the product's availability. If the product is not in stock, the salesperson will delete the customer's order. And after that, the customer will be automatically notified of the cancellation by email. If the order is complete, the storekeeper packs the order, the software generates the invoice, and the stock is changed. The customer is notified at the same time of the shipment of their order.

4.1. CIM level presentation

The CIM level was expressed in two models, namely the business process model and the BPMN diagram, the latter is drawn in Figure 1 [11], in this model tasks were simply represented to better describe and simplify the process. However, the identification of collaborators and subprocesses was avoided to just be able to describe a generic business process to make the shift from CIM to PIM easier. In the diagram, the tasks were classified into three categories, namely manual, user and service tasks, we have also identified two distinct gateways that are both parallel and exclusive. The BPMN business process diagram model, shown in Figure 2, is another model to express the CIM level, in this diagram; every task's output included a data object with its state, as well as all possible paths.

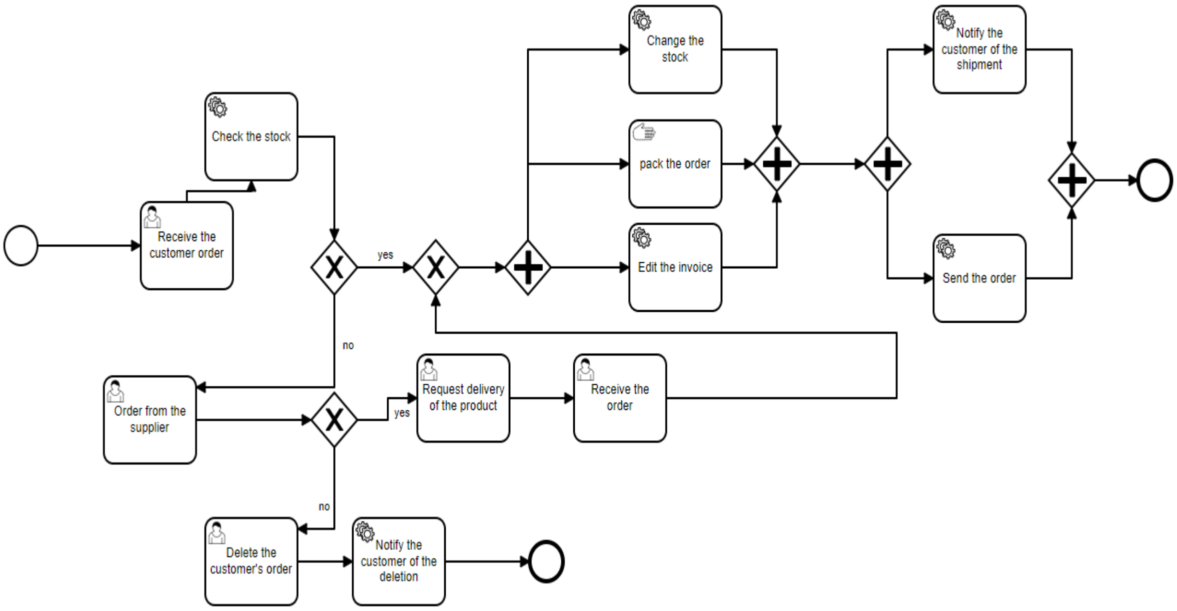

Change the stock
Notify the customer of the shipment
Check the stock
pack the order
Receive the customer order
yes
Edit the invoice
Send the order
no
Request delivery of the product
Receive the order
Order from the supplier
yes
no
Delete the customer's order
Notify the customer of the deletion

Figure 1. BPMN diagram of the order management process

4.2. Presentation of the CIM metamodel: BPMN metamodel

The BPMN metamodel is shown in Figure 3 [12]. In our approach, the business process was represented by utilizing BPMN at the CIM level. Due to the business process being computationally independent, the OMG norm for business process modeling is BPMN.

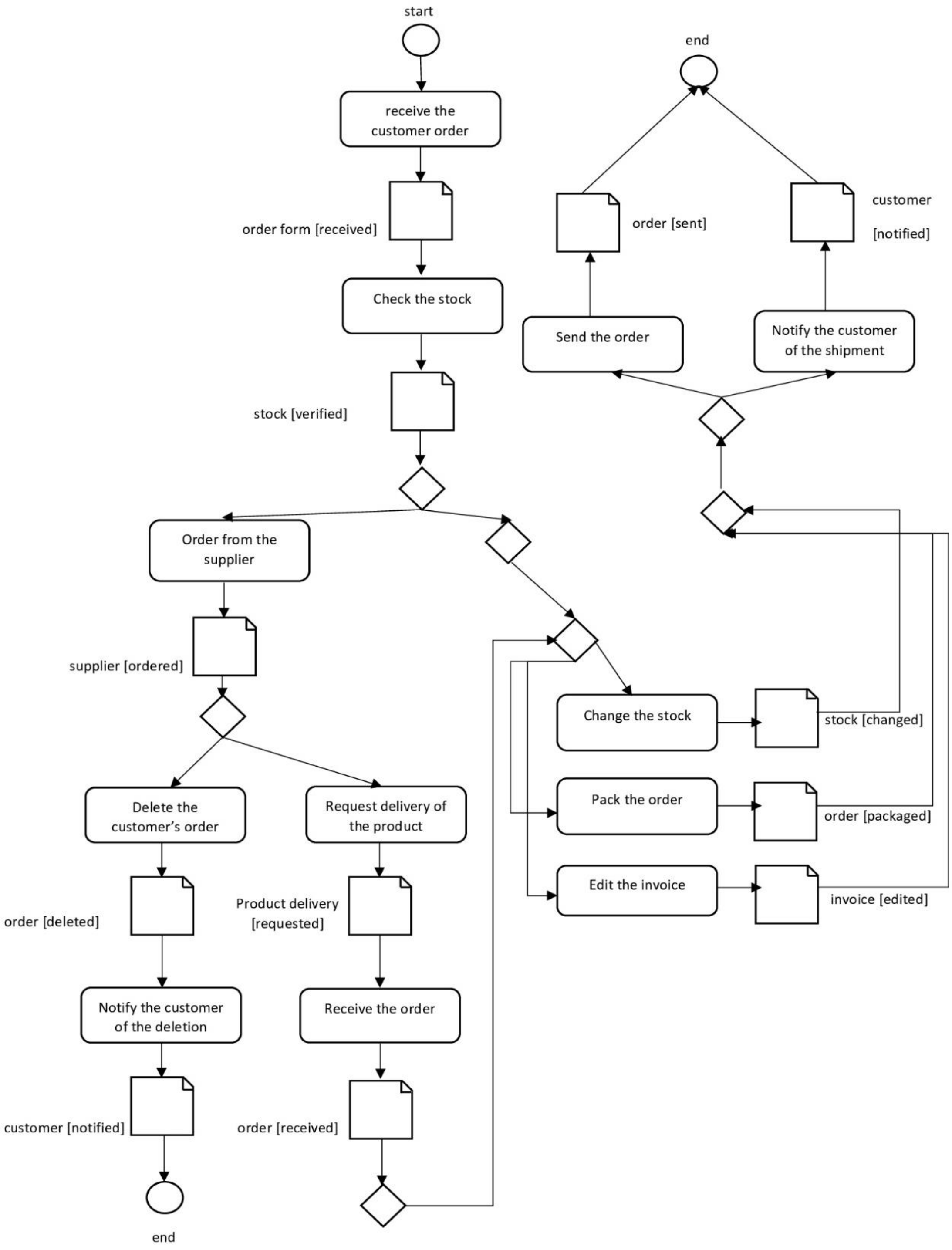


Figure 2. BPMN business process diagram of the order management process

4.3. PIM level presentation

The IFML model created by implementing the transformation rules from the CIM level represented by BPMN is illustrated in Figure 4.

4.4. Presentation of the PIM metamodel: IFML metamodel

Figure 5 [13] and Figure 6 [14] present the IFML metamodel. The latter utilizing the main data types of the UML metamodel provides a set of UML metaclasses as the foundation for IFML metaclasses and assumes that the IFML ContentModel is described in UML.

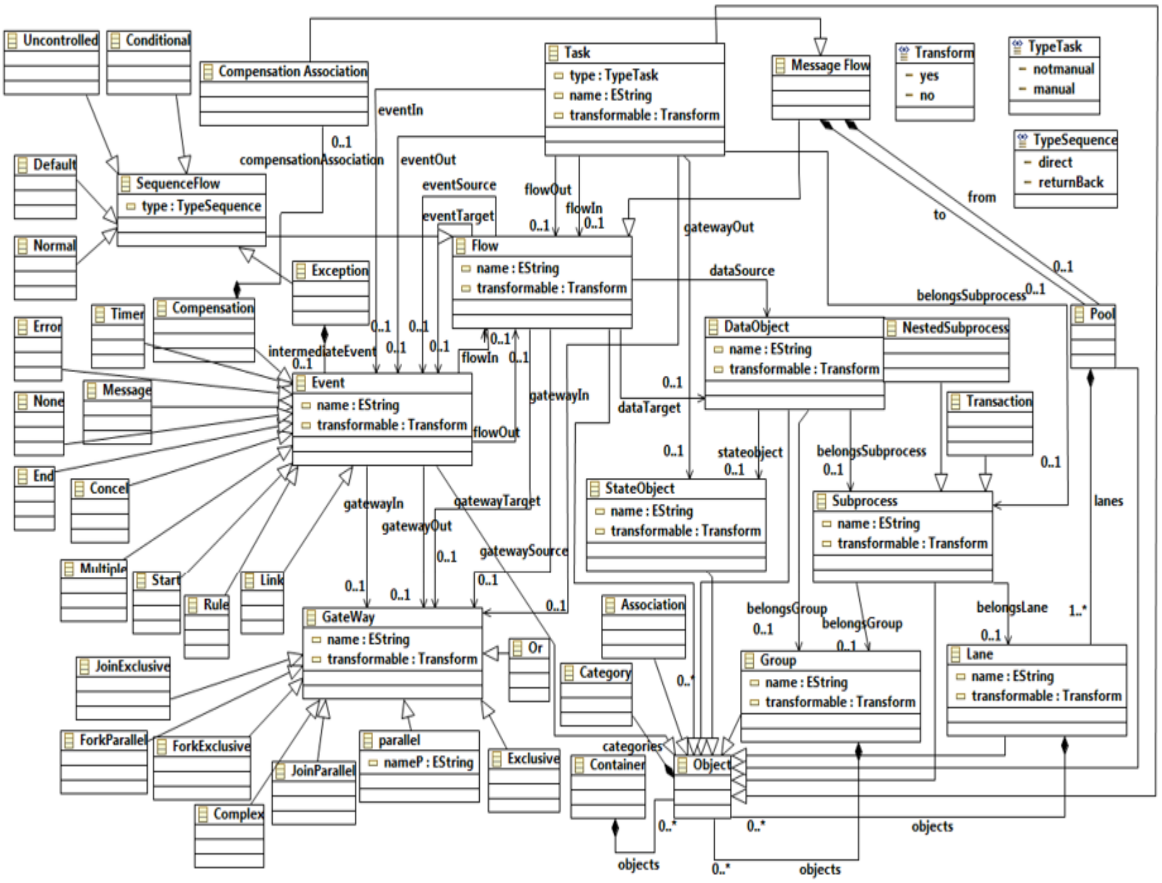


Figure 3. BPMN metamodel

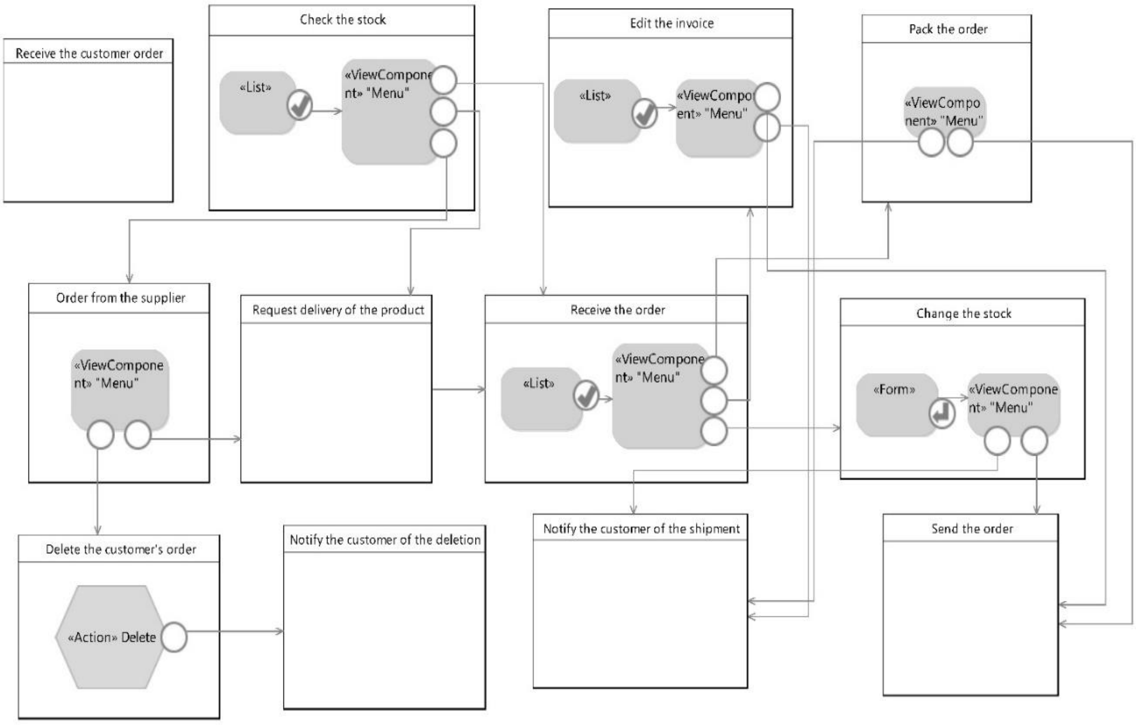


Figure 4. IFML model of the order management process

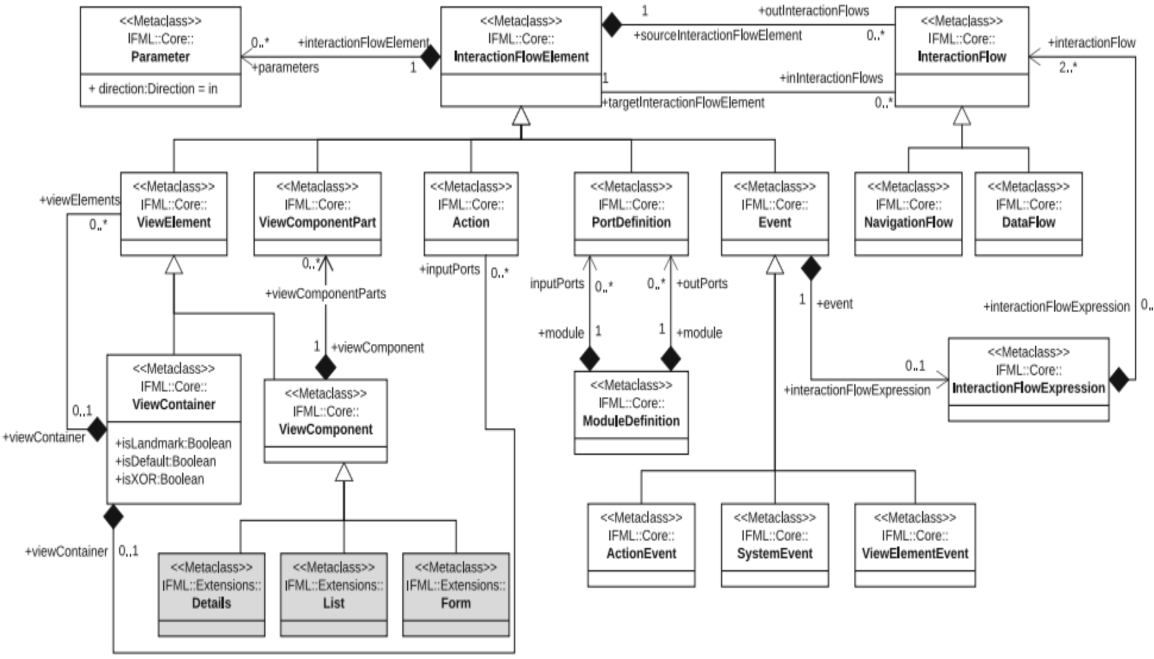


Figure 5. IFML metamodel (a)

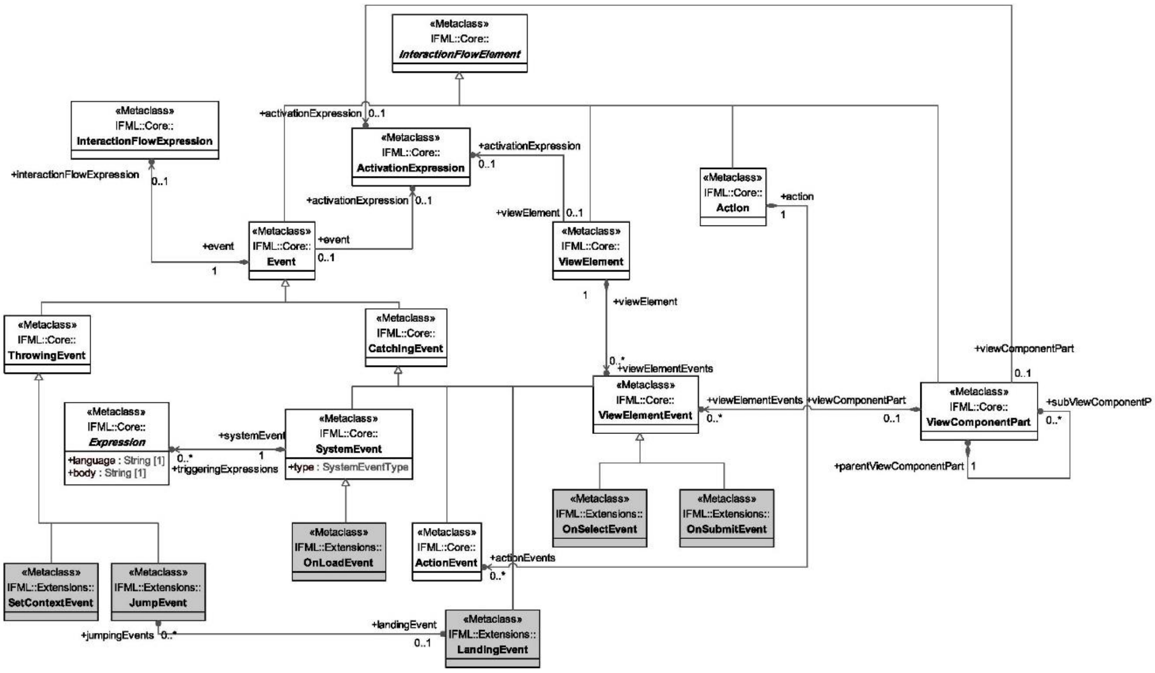


Figure 6. IFML metamodel (b)

5. DISCUSSION AND LIMITATIONS

According to the below-mentioned literature [15]-[25], All of those searches on transforming CIM to PIM have been achieved, and we never really covered the semi-automatic production of graphical user interfaces. MDA is conducting this study that uses BPMN and IFML standards and does not need much code in comparison to the conventional software development methodologies. This paper can be a guide to overcoming the problem of semi-automating the transformation from BPMN to IFML. Our methodology is defined but not yet implemented in this paper, although it can be achieved and graphical user interfaces can be developed automatically employing this approach.

6. CONCLUSION AND FUTURE WORK

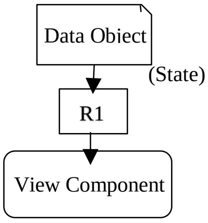
In software engineering, one of the most difficult issues is developing a method for semi-automating the transformation of business process models into software design models. MDA possesses the ability to persuade the technology industry of its importance, while also allowing for significant cost reductions in software engineering.

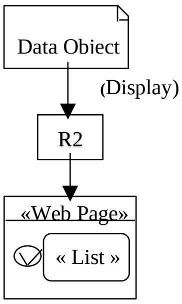
In this study, we have presented a new methodology for transforming BPMN to IFML into MDA, by utilizing the transformation rules that have been developed, the proposed methodology provides an efficient remedy for transforming the CIM level describing business models to analysis and design models presented at the PIM level. Our approach was illustrated with an example of the order management process in which the IFML model was obtained by using the shift rules from the CIM level. The elaboration of transformation rules from BPMN to IFML; allowed CIM to PIM transformation in a semi-automatic manner. Advancement aims to improve the PSM level and obtain the code automatically as well as implement the graphical user interfaces gained through applying the CIM rules shift to the PIM level.

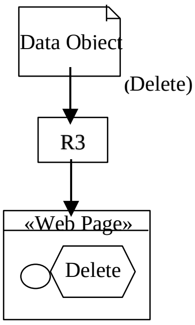
APPENDIX

Table 1. Rules transformation from BPMN models to IFML models

Human language	Graphical representation	Transformation rule in ATL

-Every “State” of “Data Object” becomes a “View Component”		Rule R 1{
		from
		stt : MMbpmn!StateObject
		(stt.isTransformableStateObject ())
		to
		vcp : MMifml!ViewComponent(
		Name <- stt.name
		) }

-Every “Display” state of “Data Object” becomes “View Container” with “List” view component plus “On Select Event”		Rule R 2{
		from
		stt : MMbpmn!StateObject
		(stt.isTransformableStateObject () and stt.isDisplay())
		to
		vcn : MMifml!ViewContainer (
		Name <- ‘«WebPage»’+stt.name)
		lst : MMifml!List(name<-‘«List»’, ContainerBelongs<-vcn.name)
		Osev : MMifml!OnSelectEvent(
		ViewComponentBelongs<-lst.name)
		}

-Each “Delete” state of “Data Object” becomes “ View Container” with a “Delete” action plus an event		Rule R 3{
		from
		stt : MMbpmn!StateObject
		(stt.isTransformableStateObject () and stt.isDelete())
		to
		vcn : MMifml!ViewContainer (
		Name <- ‘«WebPage»’+stt.name)
		ac : MMifml!Action (name<-‘Delete’, ContainerBelongs<-vcn.name)
		ev : MMifml!Event(
		ViewComponentBelongs<-ac.name)
		}

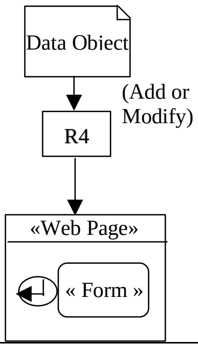
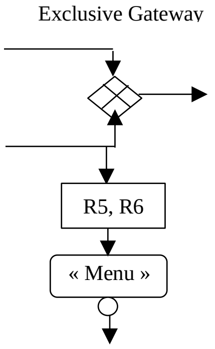
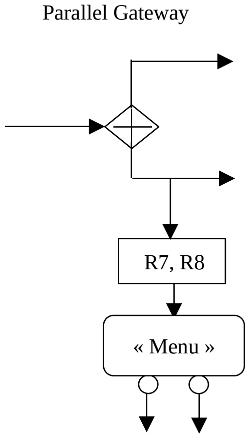
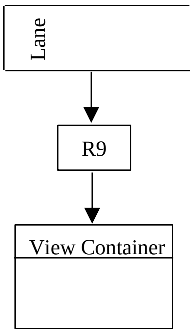
-Every “Add or Modify” state of “Data Object” becomes “View Container” with “Form” view component plus “On submit an event”	 Data Object
	(Add or Modify)
	R4
	«Web Page»
	« Form »
		Rule R 4{
		from
		stt : MMbpmn!StateObject
		(stt.isTransformableStateObject () and (stt.isAdd() or stt.isModify()))
		to
		vcn : MMifml!ViewContainer (
		name <- ‘«WebPage»’+stt.name)
		frm : MMifml!Form(
		name<-‘Form’+ stt.name, containerBelongs<-vcn.name)
		osev : MMifml!OnSubmitEvent(
		ViewComponentBelongs<-frm.name)
		}

Table 1. Rules transformation from BPMN models to IFML models (continue)

Human Language	Graphical representation	Transformation rule in ATL

-Every “Exclusive Gateway” becomes a “Menu” view component with an event related output transition “Exclusive Gateway		Rule R 5{
		from
		exl : MMbpmn!Exclusive
		(exl.isTransformableEclusive ())
		to
		vcp: MMifml!ViewComponent (
		name<- ‘« Menu »’+exl.name)}
		Rule R 6 {
		from
		fl : MMbpmn!Flow
		(fl.isTransformableFlow() and
		fl.OutputisDecisionState())
		to
		MMifml!Event(name<-fl.name,
		ViewComponentBelongs<-
		fl.NameDecisionStateOutput())

-Every “Parallel Gateway” becomes a “Menu” view component plus “Events” related output transition “Parallel Gateway”		Rule R 7 {
		from
		prl : MMbpmn!Parallel
		(prl.isTransformableParallel ())
		to
		vcp: MMifml!ViewComponent (
		name<- ‘« Menu »’+prl.name)}
		Rule R 8 {
		from
		fl : MMbpmn!Flow
		(fl.isTransformableFlow() and
		fl.OutputisDecisionState())
		to
		MMifml!Event(name<-fl.name,
		ViewComponentBelongs<-
		fl.NameDecisionStateOutput())
		}

-Every “Lane” becomes a “View Container”		Rule R 9{
		from
		ln : MMbpmn!Lane
		(ln.isTransformableLane())
		to
		vcn : MMifml!ViewContainer (
		Name <- ln.name
		)
		}

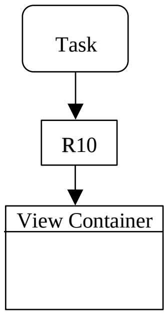
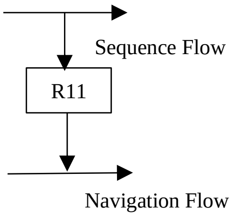
-Every “Task” becomes a “View Container”		Rule R 10{
		from
		tsk : MMbpmn!Task
		((not tsk.isManual()) and
		(tsk.isTransformableTask())
		to
		vcn : MMifml!ViewContainer (
		Name <- tsk.name
		)
		}

Table 1. Rules transformation from BPMN models to IFML models (continue)

Human Language	Human Language	Human Language

-Every "Sequence Flow" becomes a "Navigation Flow"		Rule R 11{
		from
		sqf : MMbpmn!SequenceFlow
		(sqf.isTransformableSequenceFlow())
		to
		nvf : MMifml!NavigationFlow (
		name <- sqf.name
		source <- sqf.source()
		target <- sqf.target()
		)
		}

REFERENCES

[1] OMG-MDA, "MDA Guide Version 1.0.1," 2003, *[Online].* Available at: https://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf/.

[2] R. R-Echeverria, J. C. Preciado, J. Sierra, J. M. Conejero, and F. Sanchez-Figueroa, "AutoCRUD: Automatic generation of CRUD specifications in interaction flow modeling language," *Science of Computer Programming*, 2018, doi: 10.1016/j.scico.2018.09.004.

[3] A. Huang, M. Pan, T. Zhang, and X. Li, "Static Extraction of IFML Models for Android Apps," *MODELS '18: Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, 2018, doi: 10.1145/3270112.3278185.

[4] F. Waheed, F. Azam, M. W. Anwar, and Y. Rasheed, "Model Driven Approach for Automatic Script Generation in Stress Testing of Web Applications," *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, April 2020, pp. 46–50, doi: 10.1145/3397125.3397137.

[5] R. R-Echeverria, J. C. Preciado, A. R-Largo, J. M. Conejero, and A. E. Prieto, "A Pattern-Based Development Approach for Interaction Flow Modeling Language," *Hindawi Scientific Programming*, vol. 2019, 2019, doi:10.1155/2019/7904353.

[6] M. Hamdani, W. H. Butt, M. W. Anwar, I. Ahsan, F. Azam and M. A. Ahmed, "A Novel Framework to Automatically Generate IFML Models From Plain Text Requirements," in *IEEE Access*, vol. 7, pp. 183489-183513, 2019, doi: 10.1109/ACCESS.2019.2959813.

[7] N. Laaz, N. Kharmoum, and S. Mbarki, "Combining Domain Ontologies and BPMN Models at the CIM Level to generate IFML Models," *Procedia Computer Science*, vol. 170, pp. 851–856, April 2020, doi: 10.1016/j.procs.2020.03.145.

[8] OMG, "Business Process Model, and Notation (BPMN)-Version 2.0," 2011, *[Online].* Available at: https://www.omg.org/spec/BPMN/2.0/PDF/?iOS=/ (accessed Feb. 26, 2022).

[9] M. Brambilla, and P. Fraternali, "Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML," *IFML-Interaction Flow Modeling Language*, London: Morgan Kaufmann, ISBN: 978-0128001080, December 2014.

[10] N. Laaz, K. Wakil, S. Mbarki and D. N. A. Jawawi, "Comparative Analysis of Interaction Flow Modeling Language Tools," *Journal of Computer Science*, vol. 14, no. 9, pp. 1267-1278, 2018, doi: 10.3844/jcssp.2018.1267.1278.

[11] F. Paupier, "All about business process modeling," 2017, (in French), *[Online].* Available at: https://www.appvizer.fr/magazine/operations/business-process/modelisation-processus-metier/ (accessed Mar. 01, 2022).

[12] Y. Rhazali, Y. Hadi, and A. Mouloudi, "A Methodology of Model Transformation in MDA: from CIM to PIM," *International Review on Computers and Software (I.RE.CO.S.)*, vol. 10, no. 12, pp. 1186-1201, 2015, doi: 10.15866/irecos.v10i12.8088.

[13] M. Brambilla, E. Umuhoza and R. Acerbis, "Model-driven development of user interfaces for IoT systems via domain-specific components and patterns," *Journal of Internet Services and Applications*, vol. 8, no. 14, 2017, doi:10.1186/s13174-017-0064-1.

[14] M.Brambilla, "Modeling and execution of software user interfaces," *3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, 2015, pp. IS-9.

[15] S. Gotti, S. Mbarki, Z. Gotti and N. Laaz, "A Model Driven approach for multi-platform execution of interactive UIS designed with IFML," *International Journal of Computing*, vol. 18, no. 3, pp. 293-306, 2019, doi: 10.47839/ijc.18.3.1522.

[16] 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)*, 2015, pp. 1-6, doi: 10.1109/ICoCS.2015.7483318.

[17] 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.

[18] 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, 2016, pp. 125-142, doi: 10.14257/ijseia.2016.10.4.13.

[19] N. Silega, M. Noguera and D. Macias, "Ontology-based Transformation from CIM to PIM," in *IEEE Latin America Transactions*, vol. 14, no. 9, pp. 4156-4165, Sept. 2016, doi: 10.1109/TLA.2016.7785947.

[20] L. Zonghua, Z. Xiaofeng, and Y. Zhengwei, "A Formalization Model Transformation Approach on Workflow Automatic Execution from CIM Level to PIM Level," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 9, pp. 1179–1217, 2019, doi : 10.1142/S0218194019500372.

[21] V. D. Castro, E. Marcos and J. M. Vara, "Applying CIM-to-PIM model transformations for the service-oriented development of information systems", *Information and Software Technology*, vol. 53, no. 1, pp. 87-105, 2011, doi: 10.1016/j.infsof.2010.09.002.

[22] N. Kharmoum, S. Ziti, Y. Rhazali, and F. Omary, "An automatic transformation method from the E3value model to IFML Model: An MDA approach," *Journal of Computer Science*, vol. 15, no. 6, pp. 800-813, 2019, doi: 10.3844/jcssp.2019.800.813

[23] I. Essebaa and S. Chantit, "QVT Transformation Rules to Get PIM Model from CIM Model*", Advances in intelligent system and computing*, vol. 520, pp. 195-207, 2017, doi: 10.1007/978-3-319-46568-5_20.

[24] 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, pp. 170-181, 2018, doi: 10.3991/IJOE.V14I09.9137.

[25] A. Kriouile, N. Addamssiri and T. Gadi, "An MDA Method for Automatic Transformation of Models from CIM to PIM," *American Journal of Software Engineering and Applications*, vol. 4, no. 1, pp. 1-14, 2015, doi: 10.11648/j.ajsea.20150401.11.

BIOGRAPHIES OF AUTHORS

Abir Sajji iD 8 SC P Ibn Tofail University awarded her a Master's Degree in Software Quality at the Faculty of Sciences Kenitra, Morocco, in 2010. She after worked at the presidency of the same university as an administrator, currently, she prepares her Ph.D. degree in the Ibn Tofail University, Faculty of Sciences Kenitra, Morocco, in 2021. Her primary research topics are software processes and conceptual modeling. She can be contacted at email: abir.sajji@uit.ac.ma.

Yassine Rhazali iD 8 SC P is a Moroccan professor, he receive his Ph.D. in computer science at Ibn Tofail University, Kenitra, Morocco, more specifically in software engineering, his specialty is the Model Driven Engineering, he proposed several approaches in model engineering, and his approaches are validated in various scientific papers published at international scientific conferences and international scientific journals. He is now a professor-researcher, reviewer, editorial board member, and scientific committee member in several international scientific journals and at various international scientific conferences. He has taught computer science in varied universities and institutes. He can be contacted at email: dr.yassine.rhazali@gmail.com.

Youssef Hadi iD 8 SC P In 2008, he obtained his Ph.D. from Mohammed V University's Faculty of Sciences in Rabat, Morocco. He is now an Assistant Director for Educational Affairs at the Higher School of Technology (EST) in Kenitra, Morocco, where he is in charge of many Master's Formations sections. He is the director of software engineering and multimedia database indexing research teams. In such areas of research, he has made significant contributions and publications. He can be contacted at email: hadiyoussef@gmail.com.