IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 July 2022, accepted 12 August 2022, date of publication 25 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201556

TOPICAL REVIEW

Transformation From CIM to PIM: A Systematic Mapping

NEMURY SILEGA[1], MANUEL NOGUERA[2], YURI I. ROGOZOV[1], VYACHESLAV S. LAPSHIN[1], AND TAHUMARA GONZÁLEZ[3]

[1]Department of System Analysis and Telecommunications, Southern Federal University, 347900 Taganrog, Russia
[2]Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, 18071 Granada, Spain
[3]Centro de Informática Industrial, Facultad 4, Universidad de las Ciencias Informáticas, Habana 19370, Cuba

Corresponding author: Nemury Silega (silega@sfedu.ru)

This work was supported by the Postdoctoral Fellowship through the Institute of Computer Technologies and Information Security, Southern Federal University, under Project PD/20-02-KT.

ABSTRACT Model Driven Architecture (MDA) is the most prominent and accepted methodology based on the Model Driven Development (MDD) principles. MDA includes three abstraction levels: Computer Independent Models (CIM), Platform Independent models (PIM) and Platform specific models (PSM). MDA encourages the automatic transformation of models as a means to increase the speed of the software development process and to prevent human errors. There are plenty of solutions to transform PIMs to PSMs, however the CIM to PIM transformation does not receive a similar attention. In that sense, this paper aims to describe a systematic mapping to analyze the main characteristics of the approaches that deal with the CIM to PIM transformation as well as to discuss research directions stemming out from our analysis. The results of this mapping study could be a valuable information source for the scientific community in order to know the real advances in this topic and to avoid unnecessary effort dealing with problems that have already been addressed. For example, this study yielded the models at the CIM level that have already been transformed into models at the PIM level. Hence, with this information, the researchers could focus their attention on finding solutions to transform those models at CIM level that have not been transformed into models at PIM level. Likewise, this mapping study provides information regarding the technological support of this type of transformation. This information could be useful for those software projects interested to adopt MDA.

INDEX TERMS Model driven architecture (MDA), computer independent models (CIM), platform independent models (PIM), systematic mapping.

I. INTRODUCTION

Model Driven Architecture (MDA) is a software development methodology based on the Model Driven Development (MDD) promoted by the Object Management Group (OMG). Defining the structure, semantics, and notations of models using industry standards is the main specification of MDA which enables us to deal with complexity and derive value from models and modeling [1].

MDA encourages the creation and transformation of models at different abstraction levels. Three types of models are defined: Computation Independent Models (CIMs), Platform Independent Models (PIMs) and Platform Specific Models (PSMs). As models increase the abstraction level on enterprise software systems, they improve the communication, understanding and analysis between software developers [2].

The adoption of MDA specifications may increase the productivity in the software development process, reducing costs in terms of implementation time and supporting the evolution of the systems. Since these advantages, several solutions for the transformations Model to Model (M2M) have been developed, generally from PIM to PSM and from PSM to Code. However, it is not easy to find approaches that deal with the CIM to PIM transformation [3]. This gap hinders the exploitation of the MDD capabilities for the whole software lifecycle. For example, it is not possible to ensure that PIM models cover completely all the elements captured in the business models. In spite of these facts, the

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

TABLE 1. Criteria applied in previous reviews.

Criteria	[11]	[12]	[13]	Our review
CIM Coverage	√	√	√	√
PIM Coverage	√	√	X	√
Transformation languages	X	X	√	√
Tool support	√	√	√	√
Type of transformation (Source-target)	√	X	X	√
Impact assessment	X	X	X	√
Research type	√	X	√	√
Methodological concerns	X	X	X	√
Traceability support	√	X	X	X
Number of analyzed works	4	63	*	53
Year of publication	2012	2019	2020	2022

√ - The criterion is considered
X- The criterion is not considered
*- We did not find the information regarding the number of analyzed works

research community does not stop at finding solutions for the transformation of CIM to PIM, even some of them have been able to automate some transformations. The transformation from CIM to PIM may avoid misunderstandings in the early stages of development, being important to improve the communication between business experts, stakeholders, software analysts and software designers.

On the other hand, the researchers who are interested to deal with the CIM to PIM transformation have to devote a considerable time to gather and analyze the published materials related to this topic. Usually, this type of information is presented by review studies in a compressible and resumed format. Although several reviews related to MDD have been carried out [3], [4], [5], [6], [7], [8], [9], [10], we found few reviews regarding this topic. As the best of our knowledge, the study of Sharifi *et al.* [11], is the only review addressing the CIM to PIM transformation. However, since this study was conducted in 2012, it is suitable to analyze the most recent advances in this field. The criteria applied by Hamid Reza Sharifi *et al.* were used as a guide to define the criteria for our review.

Habba *et al.* analyzed approaches that discuss the operational alignment by reducing the gap between business requirement, business process, and software system [12]. However, they are not focused on the MDD context. Hence, relevant questions like transformation languages and transformation tools are beyond their scope. Moreover, this review does not explore the methodological dimension of the approaches or their impact in improving the software development process.

The review of Kharmoum *et al.* is focused only on the CIM level but includes some criteria that we reused for our mapping, for example the criteria regarding the models construction and transformation [13]. Table 1 depicts a summary table with the main criteria that these reviews consider.

Taking into account the aforementioned reasons, the aim of this paper is to characterize the approaches that deal with the transformation from CIM to PIM. To conduct the review of the existing proposals, we have applied the method of systematic mapping. This method allows to make a characterization of the revised works taking into account the fulfillment of a set of steps that lead to the results that can be classified [14]. The results of this method will show the progress in this topic as well as the main gaps that may be addressed for the scientific community.

The results of this review could be useful for both researchers and software developers. The researchers on MDA field can either deal with the gaps of the existing transformations or create new transformations. Whereas software developers can adopt for their projects those CIMs and PIMS which have been employed in automatic transformations. Likewise, this mapping study attempts to clarify regarding the technological support for this type of transformation. Hence, the software development managers can exploit this information to choose the most suitable technology according to the characteristics of their projects.

Furthermore, this study will illuminate about the real state of the adoption of formal models in the CIM and PIM levels. Formal models are a collection of well-defined mathematically-based techniques [15]. A formal modelling language consists of a description of its well-defined syntax and semantics that enhance the readability and the expressiveness of the language. This type of models can be used to check system properties such as performance, reachability, consistency and correctness, mathematically [15]. The negative consequences of adopting non-formal notations have been extensively discussed for the scientific community [16]. Likewise, the benefits of adopting formal models in the MDD context have been discussed in the works which are in the intersection of software engineering community and ontological engineering community [17], [18]. Hence, analyzing the real state of this topic might yield valuable insights for both communities.

On the other hand, to know what methods have been applied to evaluate the transformations is a concern for the scientific community. Hence, we also attempt to provide the answer to that question.

Likewise, usually the MDD-based approaches lack of methodological specifications. This gap may hinder the application of these approaches in a systematic and unambiguous manner. Therefore, to know the real state about this issue in the approaches that deal with the CIM to PIM transformation could be useful to confirm if this is an existent problem.

The rest of the paper has been structured as follows. In Section II the process of systematic mapping is explained. In Section III the results are analyzed. Finally, the conclusions and related work are presented.

II. MODEL DRIVEN ARCHITECTURE (MDA)

MDA lays on four principles [19], [20]:

1) Models expressed in a well-defined notation are a cornerstone to system understanding for enterprise-scale solutions.

2) Building systems can be organized around a set of models by imposing a series of transformations between models, organized into an architectural framework of layers and transformations.
3) A formal underpinning for describing models in a set of metamodels facilitates meaningful integration and transformation among models, and is the basis for automation through tools.
4) Acceptance and broad adoption of this model-based approach requires industry standards to provide openness to consumers, and foster competition among vendors.

Some ways to derive value from models and modeling that contribute dealing with the complexity and interdependence of complex systems are [1]: Models as communications vehicles; derivation via automated transformation; model analytics; model simulation and execution; deriving information from models; and structuring unstructured Information.

Three types of models are defined in MDA: Computation Independent Models (CIMs), Platform Independent Models (PIMs) and Platform Specific Models (PSMs). A CIM does not show details of the structure of systems. A CIM is frequently referred to as a domain model, and its specification uses a vocabulary that is common to users of the domain area. A PIM describes the system, but does not show details of its use of its platform. On other hand, a PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform [2].

CIMs are crucial in bridging the gap between the domain experts and the experts of the design and construction of the artifacts that together fulfill the domain requirements. CIMs help in presenting exactly what the system is expected to do. CIMs are valuable, not only to comprehend a problem, but also as a instrument to share vocabulary with other models [21].

The CIM to PIM transformation allows to use the CIMs not only as a means of communication between domain experts and software developers. This transformation is a key step to ensure that business information is conveyed and respected throughout the MDA process [22]. In addition, this transformation enables traceability links between requirement specifications, represented in the CIMs, and PIM and PSM artefacts that implement them [21]. This link contributes to ensure the quality in the software development process by making easier that any changes in CIM level will reflect the PIM and the PSM levels [23].

III. METHOD OF SYSTEMATIC MAPPING

A systematic mapping (SM) is a secondary study to create a classification scheme and structure an interesting field in software engineering [14]. The SM results enable the proposals and results of research into a certain field to be organized by thematically analyzing and identifying the main outlets of publication. An SM is considered a prior and necessary step for determining suitable areas for a more in-depth study [24] and conceives the follow stages [25]:

1. Definition of research questions (RQ)
2. Search execution
3. Work selection and quality evaluation
4. Data extraction
5. Analysis and classification
6. Mapping evaluation

The following sections present the results of the activities for each stage in our mapping.

A. DEFINITION OF RESEARCH QUESTIONS (RQ)

RQs should be defined according to the main study objectives and should consider four elements:

1. **Population**: researchers and professionals involved in the creation or adoption of MDA-based approaches.
2. **Study factor**: current state of the transformation from CIM to PIM.
3. **Intervention in the comparison**: approaches that deal with the transformation from CIM to PIM.
4. **Result**: main features of the approaches that deal with the transformation from CIM to PIM.

These elements enabled the RQs to be defined. According to Petersen's recommendations, the research questions can be classified into two categories: those that focus on common elements and those that focus on the topic area. RQ.7 and RQ.8, therefore, belong to the first category whereas the other questions are focused on the study topic. We formulated the following questions for this study:

- **RQ.1** - What languages/notations are used to represent the CIM level?
- **RQ.2** - What languages/notations are used to represent the PIM level?
- **RQ.3** - What transformation languages are used?
- **RQ.4** - What tools are used to transform CIM to PIM?
- **RQ.5** - How is the transformation carried out: tool-support or manual?
- **RQ.6** – What transformations (source model- target model) have been carried out?
- **RQ.7** - What research types and research methods have been applied?
- **RQ.8:**What quality attributes of the software development process are improved with the CIM to PIM transformations?
- **RQ.9** What methodological aspects do these approaches deal with?

The answer of these research questions will help to identify the state of the art regarding the transformation from CIM to PIM during the period between the years 2002 and 2022. The starting point is 2002 because in that year OMG introduced to MDA as an approach in Software Engineering.

B. SEARCH EXECUTION

The selection of locations and the definition of the search strategy are crucial factors for a successful search. For the purposes of our study, we selected Google Scholar, ACM, IEEE, Springer and Scopus databases. Springer, Scopus,

TABLE 2. Search results.

	Google Scholar	ACM	IEEE	Scopus	Total
Search results	951	36	35	229	3273
Analyzed works	109	1	18	63	191
Candidate works	55	1	15	40	111
Selected works	24	1	11	17	53

ACM and IEEE are some of the most important databases in the field of software engineering while Google Scholar is a very popular search engine in the research community. Although Google Scholar includes ACM, IEEE, Springer and Scopus references, it also recovers documents that are not included in previous databases and which might be useful for our study. Google Scholar has been extensively applied to execute searches for systematic mappings [26].

The search string was created by combining keywords related to the research topic. This was an iterative process. We used the following search string for the searches:

"CIM and PIM" or "CIM to PIM" or "Computation independent model to Platform independent model"

For each database, we executed the search in *keywords*, *title* and *abstract* fields. We conducted the search in the period from 2002 until March, 2022.

IEEE ("All Metadata":CIM to PIM) OR ("All Metadata":"computation independent model" to "platform independent model")

Table 2 depicts the results of the search in the aforementioned databases. All these proposals were analyzed, to carry out the studies filtering (See Table 2).

C. SELECTION PROCESS

Article selection was guided by the following exclusion/inclusion criteria:

- **Inclusion**: Articles in journals, scientific events that propose solutions to the transformation from CIM to PIM in MDA environments.
- **Exclusion**: (i) Investigations that do not deal with the transformation from CIM to PIM. (ii) Investigations that present only ideas or reflections about the transformation from CIM to PIM.

The selection process included four steps. In the first step, each reviewer applied the inclusion/exclusion criteria based on the title and abstract of randomly appointed publications. The second step consisted in applying the inclusion/exclusion criteria to the introduction, conclusions and references of the remaining papers. This new set of papers is called the candidate set. Finally, the candidate papers were analyzed by a shallow reading first and with a full reading for those that raised doubts. Table 2 shows the candidate and selected papers. Finally, the execution of these four stages allowed the selection of 53 relevant works, showed in Table 3.

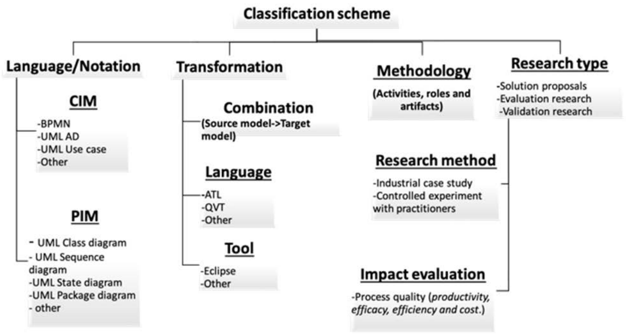


FIGURE 1. Classification scheme.

D. RELIABILITY EVALUATION OF THE INCLUSION/EXCLUSION CRITERIA

In order to evaluate the reliability of the exclusion/inclusion criteria, we selected a sub-sample of the full population of the publications. More specifically, we selected the first 50 references obtained from the Scopus search.

After applying the inclusion/exclusion criteria, the titles and abstracts of these 50 papers were used to classify the papers in a separate review (see Table 4). This review was performed by two independent reviewers. Cohen's Kappa coefficient [27] was applied to calculate inter-reviewer reliability. The reliability value was satisfactory (KA=0.71).

This result proves that there is a set of criteria which are clearly defined and that there is no significant divergence among reviewers [28].

E. CLASSIFICATION SCHEME

The classification scheme was defined according to the objectives of this study. Fig. 1 depicts a graphical representation of the classification scheme.

1) PROCESS QUALITY

This refers to the quality attributes of the development process that are improved with the analyzed approaches. The attributes are classified into productivity, efficacy, efficiency and cost.

2) METHODOLOGICAL PROPOSAL

This refers to the publications which mainly deal with the methodological aspects of a proposal. The possible values may be either *Total* (if the approach completely defines the required activities, roles and artifacts) or *Partial* (if the approach partially defines the required activities, roles and artifacts).

3) MODELS AND LANGUAGES/NOTATIONS TO REPRESENT THE CIM LEVEL

The most common models are the BPMN models, UML activity diagrams, UML Use case diagram and IDEF models.

TABLE 3. List of selected works.

Title	Authors	Year	CIMs	PIMs
From Web Requirements to Navigational Design – A Transformational Approach	Pedro Valderas, Joan Fons, Vicente Pelechano	2005	UML Activiy diagram, Attribute Graph Grammar (AGG) (XML) - Task Description	Attribute Graph Grammar (AGG) (XML) - Navigation model
Transformation from CIM to PIM: A Feature-Oriented Component-Based Approach	Wei Zhang, Hong Mei, Haiyan Zhao, Jie Yang	2005	Feature model	Conceptual component model
Towards CIM to PIM Transformation From Secure Business Processes Defined in BPMN to Use-Cases	Alfonso Rodríguez, Eduardo Fernández-Medina, Mario Piattini	2007	BPMN-BPD	UML Package diagram, UML Use case diagram, UML State diagram
CIM to PIM Transformation: A Reality	Alfonso Rodríguez, Eduardo Fernández-Medina, Mario Piattini	2008	UML (Activiy Diagram)/BPSec-Profile - Secure Business Process Model	UML Use case diagram, UML Class diagram
Transformation From CIM to PIM Using Patterns and Archetypes	Samir Kherraf, Éric Lefebvre, Witold Suryn	2008	UML Activity diagram, UML Use case diagram	UML Component model, UML Class diagram
Transforming Business Requirements into BPEL: a MDA-Based Approach to Web Application Development	Li Zhang, Wei Jiang	2008	Models of VIP Business Modeling Framework (BVM (Business Value Model), BIM (Business Information Model), BPM (Business Process Model))	BPEL model
Applying CIM-to-PIM model transformations for the service-oriented development of information systems	Valeria De Castro, Esperanza Marcos, Juan Manuel Vara	2010	BPMN-BPD, e3value Value Model	UML- Use case diagram, UML (Activiy Diagram) - Service Process Model, UML (Activiy Diagram) - Service Composition Model
An MDA Approach for Goal-oriented Requirement Analysis in Web Engineering	José Alfonso Aguilar, Irene Garrigós, Jose-Norberto Mazón, Juan Trujillo	2010	i* framework Strategic dependency (SD) model, i* framework Strategic rationale (SR) model	A-OOH- Domain model, A-OOH-Navegational model
Analytical Method of CIM to PIM Transformation in Model Driven Architecture (MDA)	Martin Kardoš, Matilda Drozdová	2010	Data Flow Diagram (DFD)	UML Use case diagram, UML Activity diagram, UML Sequence diagram, UML Domain model
Secure business process model specification through a UML 2.0 activity diagram profile	Alfonso Rodríguez, Eduardo Fernández-Medina, Juan Trujillo, Mario Piattini	2011	BPSec: activity diagrams UML 2.0 profile	UML Class diagram, UML Use case diagram
A methodology for CIM modelling and its transformation to PIM	Brahim Bousetta, Omar El Beggar, Taoufiq Gadi	2013	BPMN-BPD, UML Use case diagram	UML Class diagram, UML Sequence diagram
Deriving Multi-Agent System Behavior	Ahmed Harbouche, Mohammed Erradi, Aicha Mokhtari	2013	UML Activity diagram	UML State machine
Model Transformation Recommendations for Service-Oriented Architectures	Dmitri Valeri Panfilenko, Katsiaryna Hrom, Brian Elvesæter , Einar Landre	2013	BPMN-BPD	SoaML model (three views)
Research on Model Transformation Method Oriented to Model Driven ERP System	Y. Che, G. Wang, M. Lv, B.Y. Ren	2013	OWL Business process model	OWL Workflow model
Generating the PIM Behavioral Model from the CIM using QVT	Najiba Addamssiri, Abdelouhaed Kriouile, Youssef Balouki, Gadi Taoufiq	2014	BPMN-BPD, UML Use case diagram	UML - Sequence diagram
Getting the Static Model of PIM from the CIM	Najiba Addamssiri, Abdelouhaed Kriouile, Gadi Taoufiq	2014	BPMN-BPD, UML Use case diagram	UML Class diagram
Obtaining Behavioral Model of PIM from the CIM	Abdelouhaed Kriouile, Gadi Taoufiq, El Khadimi Ahmed	2014	BPMN-BPD, UML Use case diagram	UML Sequence diagram
Transformation-Method-CIM-to-PIM-From-Business-Processes-Models-Defined-in-BPMN-to-Use-Case-and-Class-Models-Defined-in-UML	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2014	BPMN-BPD, BPMN-CD	UML Use case diagram, UML Class diagram
An MDA Method for Automatic Transformation of Models from CIM to PIM	Abdelouahed Kriouile, Najiba Addamssiri, Taoufiq Gadi	2015	BPMN-BPD, UML Use case diagram	UML Class diagram, UML Sequence diagram

TABLE 3. *(Continued.)* List of selected works.

Disciplined approach for transformation CIM to PIM in MDA	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2015	UML Activity diagram, BPMN-CD	UML Class diagram, UML Package diagram, UML Use case diagram, UML State diagram
Diseñando Transformaciones de Modelos CIM PIM desde un enfoque de negocio hacia un enfoque de sistema	Cecilia Ariste, Julieta Ponisio, Leopoldo Nahuel, Roxana Giandini	2015	BPMN-BPD	UML Activity diagram
Transformation Approach CIM to PIM from business process models to state machine and package models	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2015	UMLActivity diagram, BPMN-CD	UML Class diagram, UML State diagram
A Methodology of Model Transformation in MDA: from CIM to PIM	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2015	BPMN-BPD	UML Package diagram, UML Use case diagram, UML State diagram, UML Class diagram
A Model Transformation in MDA from CIM to PIM Represented by web models through SoaML and IFML	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2016	UML Activity diagram	UML Class diagram, UML Use case diagram, UML State diagram, UML Class diagram, UML Package diagram
A New Methodology CIM to PIM Transformation Resulting from an Analytical Survey	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2016	BPMN-BPD	UML Class diagram, UML Use case diagram, UML State diagram
CIM to PIM Transformation in MDA from Service oriented business models to web-based design models	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2016	SoaML service choreography (UML Activity diagram), SoaML service architecture	UML Use case diagram, UML State diagram, UML Class diagram, UML Package diagram, Web modeling diagram, SOAML Software components model
Model Transformation with ATL into MDA from CIM to PIM Structured through MVC	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2016	BPMN-BPD, BPMN-CD	UML Class diagram, UML Package diagram, UML Use case diagram, UML State diagram, UML-Class diagram (MVC structured)
Ontology-based Transformation from CIM to PIM	N. Silega, M. Noguera, D. Macías	2016	BPMN-BPD, OWL Business process model	UML Component model, OWL Component model
Toward an automatic approach to get PIM level from CIM level using QVT rules	Imane Essebaa, Salima Chantit	2016	UML Activity diagram, UML Use case diagram	UML class diagram, UML sequence diagram
A based rule method to transform CIM to PIM in MDA	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2016	BPMN-BPD, UML activity diagram	UML Package diagram, UML Use case diagram, UML State diagram, UML class diagram
Model transformation from CIM to PIM in MDA from business models defined in DFD to design models defined in UML	Yassine Rhazali, Youssef Hadi, Abdelaziz Mouloudi	2016	UML Activity diagram, Data flow diagram	UML class diagram, UML package diagram, UML use case diagram, UML state diagram
Applying a Model Driven Architecture Approach: Transforming CIM to PIM Using UML	Oualid Betari!, Saida Filali, Amine Azzaoui, Mohamed Amine Boubnad	2018	UML Use case diagram	UML class diagram
Applying CIM-to-PIM Model Transformation for Development of Emotional Intelligence Tests Platform	Karim Arrhioui, Samir Mbarki, Mohammed Erramdani	2018	Emotional Model (Ecore metamodel)	UML Class diagram
A Model Transformation in Model Driven Architecture from Business Model to Web Model	Yassine Rhazali, Youssef Hadi, Idriss Chana, Mohammed Lahmer, and Abdallah Rhattoy	2018	UML Activity diagram	UML use case diagram, UML state diagram, UML class diagram, UML package diagram
Automatic Transformation of User Stories into UML Use Case Diagrams using NLP Techniques	Meryem Elallaouia, Khalid Nafilb, Raja Touahni	2018	User stories (Natural language)	UML use case diagram
Designing rule-based expert systems with the aid of the model-driven development approach	Aleksandr Yurievich Yurin, Nikita Olegovich Dorodnykh ,Olga	2018	OWL Description of rule-based ESs, OWL Domain model	UML Class diagram

TABLE 3. *(Continued.)* List of selected works.

	Anatolievna Nikolaychuk, Maksim Andreevich Grishenko			
Designing BP-IS Aligned Models: An MDA-based Transformation Methodology	Wiem Khlif, Nourchene Elleuch, Enaam Alotabi and Hanêne Ben-Abdallah	2018	BPMN-BPD	UML-Class diagram
An Automatic Transformation Method from the E3value Model to IFML Model: An MDA Approach	Nassim Kharmoum, Soumia Ziti, Yassine Rhazali , Fouzia Omary	2019	e3value Value model	IFML (Ecore Metamodel)

TABLE 4. Application of the kappa index.

		R2		TOTAL
		+	-	
R1	+	26	4	30
	-	3	17	20
TOTAL		29	21	**50**

Regarding formal notations, OWL ontologies are the most common.

4) MODELS AND LANGUAGES/NOTATIONS TO REPRESENT THE PIM LEVEL

The most common models are UML Use case diagram, UML Class diagram and other UM-based models. Similar to the CIM level, regarding formal notations, OWL ontologies are the most common.

5) TRANSFORMATIONS

First of all, we are interested to know how the transformation is carried out, the values could be either manual, automatic or semiautomatic. Besides, we would like to know the transformation languages as well as the tools that support the transformations. The most popular transformation languages are ATL and QVT whereas Eclipse Modeling Framework and Eclipse Plugins are the most common technologies to support the transformations.

6) RESEARCH TYPE AND RESEARCH METHOD

This classification is based on the scheme described by Petersen [25] and which is based on the work by Wieringa *et al.* [29]. Research can either be classified as solution proposals, evaluation research or validation research. In evaluation research, it is possible to use either the industrial case study or the controlled experiment with practitioners research methods, whilst in validation research it is possible to apply either the academic case study or the laboratory experiment as the research method.

F. VALIDITY THREATS

In order to reduce the effect of potential threats, we performed the following mitigation actions:

1) DEFINITION OF RESEARCH QUESTIONS

We performed two main actions in order to cover the maximum number of aspects in our mapping study. First, we held separate brainstorming sessions with every author in order to define and cast the research questions. Additionally, the definition of the research questions followed an iterative process whereby we updated each research question and even include new ones when necessary.

2) IDENTIFICATION OF CANDIDATE STUDIES

Since software engineering keywords are not standardized, it is possible that the selected keywords in conjunction with the search string, might not retrieve every relevant publication on the topic of the mapping study, i.e., CIM to PIM transformations. In order to mitigate this risk, we also applied an improvement of the snowball sampling technique and which consisted in also checking the publications that cited some of the works already present in the first set of selected papers, but which had not been displayed by the list of search results of the aforementioned search string. The benefit of this action is twofold. On the one hand, it we discovered that the citing paper addressed CIM to PIM transformations, its inclusion in the list of papers to be analyzed permitted to broaden the mapping the study and complement it with papers which might have gone unnoticed otherwise. On the other hand, the fact that a paper on the CIM to PIM transformation topic cited a paper shortlisted in the search based on the selected keywords and the search string, served to reinforce the inclusion of the cited work in the mapping study.

3) SELECTION OF THE PRIMARY STUDIES

In order to avoid bias associated with the reviewer when selecting articles for the mapping study, the inclusion/exclusion criteria were defined. These criteria were evaluated (see Section 2.3.1) to enhance the reliability of the results and to reduce bias.

4) DEFINITION OF THE VALUES FOR THE CLASSIFICATION CRITERIA

We adopted standard criteria whenever possible. For example, for the product quality characteristic criteria we adopted the international standard ISO/IEC 9126-1. Likewise, for the

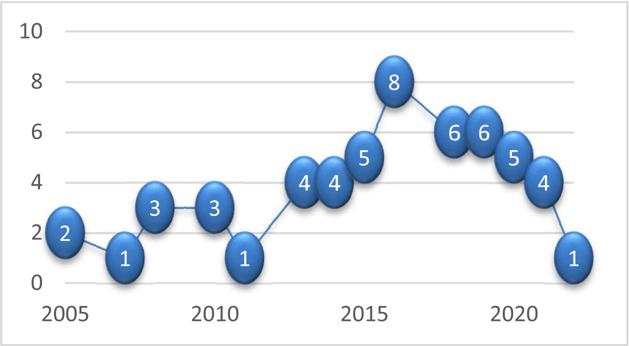


FIGURE 2. Number of publications per year.

research types and research methods we adopted the scheme proposed by Wieringa *et al.* [29].

5) DATA EXTRACTION

We created a spreadsheet to record the data of each paper. This approach made it easier to process the data. We also included a description for each criterion in order to assist the extraction process. This approach ensured that every reviewer used the same criteria. Furthermore, as some authors [30], [31] recommend, the data extracted were reviewed by a third reviewer in order to reduce bias.

IV. RESULTS AND ANALYSIS

The analysis of the results after the data extraction process is the next step in the mapping study. Fig. 2 depicts the number of publications per year in the time period 2005 until February 1, 2022. During the period 2013-2021, the number of publications in this research field was steady between 4 and 8 per year. This behavior demonstrates that the CIM to PIM transformation deserves the attention of the research community. However, it also evidences that this topic does not receive the same attention that other topics among the MDD field. For example, the number of publications related to Model driven user interface development (MDUID) has been steady above 6 publications, even with a peak of 12 in 2015 (See Fig. 3). To make a comparative analysis of the works that deal with the CIM to PIM transformation with respect to the total of works using the MDD/MDA approach, we have made a search in the SCOPUS database. We searched the papers in the subject area of *Computer science* that include "Model driven development or Model driven architecture" in title, abstract or keywords. The search yielded 5,306 documents. Hence, since we found 53 works that deal with the CIM to PIM transformation, we can estimate that these works represent about 1 % of the published works related to MDD/MDA. This is only a preliminary comparative analysis but a more rigorous, critical and comparative study of the current state of art could be an interesting matter for a future research work.

RQ.1 - *What languages/notations are used to represent the CIM level?*

TABLE 5. Most common models at CIM level.

Model	Number	%
BPMN BPD	27	51
UML Activity diagram	10	18,9
UML Use case diagram	9	17
e^{3}value-Value model	5	9,4
OWL Domain model	3	5,7
OWL Business process model	3	5,7

TABLE 6. Most common models at CIM level since 2016.

Model	Number	%
BPMN-BPD	15	50
UML Activity diagram	5	16,7
UML Use case diagram	3	10
OWL Domain model	3	10
OWL Business process model	2	6,7
e^{3}value Value model	2	6,7
User stories (Natural language)	2	6,7

TABLE 7. Most common models at CIM level before 2015.

Model	Number	%
BPMN BPD	4	28,6
UML Activity diagram	3	21,4
UML Use case diagram	2	14,3
e^{3}value-Value model	2	14,3

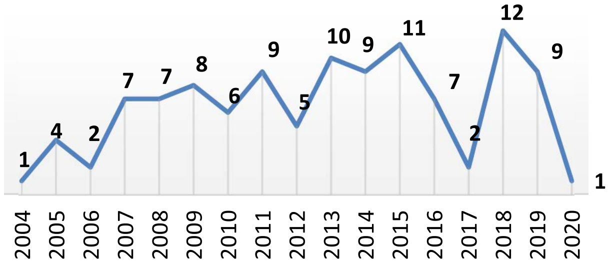


FIGURE 3. Number of publications about MDUID per year.

This is one of the most common questions that the researchers in this field address. We found that 31 types of models have been used to represent the CIM level. As we expected, UML-based models and BPMN models are the most common types of models. Table 5 shows the models that have been adopted at least for three works.

Table 1 shows the full list of the model types adopted for each analyzed work. We reduced this analysis by examining only the 30 works of the last 5 years (since 2016). This new analysis yielded that BPMN models are the most widely adopted (50%), followed by UML-based models as well (see Table 6).

To know how the adoption of these types of models has evolved, we focused the analysis on the 14 oldest approaches (that is, those produced before 2014). Similarly, to the two previous analyzes, BPMN models and UML-based models are the most frequently used (see Table 7). However, when it

TABLE 8. Formal models used at CIM level.

Model	Number
OWL Domain model	3
OWL Business process model	3
OWL Description of rule-based ESs	1
Petri Nets - Business process model	1
Attribute Graph Grammar (AGG) (XML) - Task description	1

TABLE 9. Most common models at PIM level.

Model	Number	%
UML Class diagram	32	60,4
UML Use case diagram	20	37,7
UML State diagram	11	20,8
UML Package diagram	9	17,0
UML Sequence diagram	7	13,2
IFML (Ecore metamodel)	3	5,7

TABLE 10. common models at PIM level from 2016.

Model	Number	%
UML Class diagram	22	73,3
UML Use case diagram	12	40,0
UML State diagram	7	23,3
UML Package diagram	6	20,0
IFML (Ecore metamodel)	3	10,0

comes to representing busines process models, BPMN and UML Activity diagram have been adopted almost for the same number (4 and 3 respectively) of works. This behavior demonstrates that in the past the two types of models had an equivalent usage, but currently BPMN has become the *de-facto* standard to represent business process models. The percentage of works could be a better metric to illustrate this trend. Fig. 4 clearly shows this behavior. Despite the fact that in 2016 and 2018 both types of models were used for the same number of works, it is evident that since 2010 BPMN has been the most adopted one. The fact that in several years BPMN has been used for the 100 % of the works and that since 2013 it is being adopted by, at least, 40 % of the works is solid evidence of the acceptance of BPMN.

On the other hand, to know what types of formal models have been used at the CIM level is another interesting question to address in this mapping. Regarding this topic, as it is shown in Table 8, we found that five types of formal models have been used. OWL ontologies are the most used formalism to represent business process models and domain models. The fact that only five works (5,6 % of the total) adopted formal models is strong evidence that confirms the little attention that formal models receive in this context. The analysis of causes and consequences of this finding could be a useful research direction. Most of the MDA methodologies are based on semi-formal notations and, hence, they do not enable formal reasoning about developed specifications. As it has been analyzed and formally described by other authors [16], the expressiveness of the constructs of non-formal notations (UML, BPMN, etc.) may lead to implicit consequences that can go undetected by the designer in complex diagrams, and cause various forms of inconsistencies or redundancies in the diagram itself. This may result in a degradation of the quality of the design and/or increased development times and costs. Therefore, to check that a model is consistent to its meta-model, it is required, but it is not sufficient, to ensure the quality of a model if we are not using formal notations. The main advantage of formal methods is that they offer the possibility of carrying out reasoning tasks for verification and validation purposes [32].

In the CIM level, the business is represented by the business process view and the domain view. Some authors include the system requirement view in this level as well [33]. Hence, to know the models that cover these three dimensions is another interesting question. The business process view is represented mainly by BPMN model and UML Activity diagram, whilst the requirement view is usually represented by using UML Use case diagram. These last two views are usually covered in these approaches, however few papers include domain models. OWL was the most employed formalism to represent domain models, as it was adopted in three papers [33], [34], [35]. Other models to represent domain specific information have been used, for example Emotional Model [36]. Taking into account these facts, a conclusion of this study is that the domain view has not been extensively addressed by the analyzed approaches. Since in domain models relevant busines information is identified, the insufficient coverage of this view affects the completeness of the CIM level which may hinder the automatic generation of system models.

RQ.2 -*What languages/notations are used to represent the PIM level?*

Similarly, as in the case of the CIM level, this is one of the most common questions that researchers in this field address. We found that up to 28 types of models were adopted at this level in the list works analyzed. Table 9 shows the models that have been adopted by at least 3 works. As we expected, the UML-based models are the most common. The fact that 44 works (83 % of total) adopt some UML-based is strong evidence that demonstrates the extensive use of UML to represent models at the PIM level.

We carried out an analysis of the 30 most recent works (from 2016). As Table 10 shows, the list of the most adopted models in the last years coincidences with those of the Table 9. This result demonstrates that UML is the preferred modeling language to represent the PIMs. Furthermore, this analysis reveals that IFML (Interaction Flow Modeling Language) has become in a popular modeling language. IFML is an OMG recommendation for modeling the content of front-end applications, interface composition, user interaction and control behavior [37], [38]. Likewise, the analysis of the

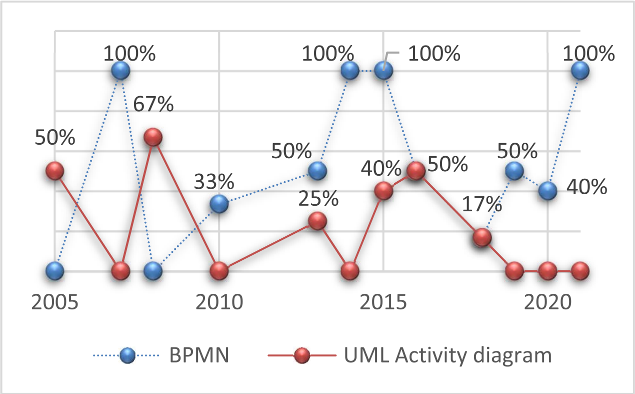


FIGURE 4. Comparison of BPMN and UML AD.

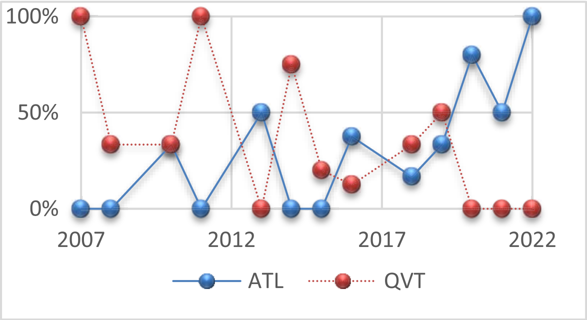


FIGURE 5. Comparison of ATL and QVT.

14 oldest works indicates that UML-based models were the most employed as well.

The fact that some models have been used in both levels is an interesting finding after analyzing the models that have been adopted to represent the CIM and PIM levels. These models are: UML Activity diagram, UML Use case diagram and UML Domain model. This finding shows that the classification of some models is still a discussion topic in this context. Specially to define the level that should be considered for the requirement models.

RQ.3 -*What transformation languages are the most adopted?*

We found that 35 works (66 %) use some transformation language. The analysis revealed that ATL (16) and QVT (14) are the most used. Hence, 86 % of the works that adopted a transformation language adopted either ATL or QVT. Only 5 works adopted a language different from ATL or QVT. The application of OWL to represent transformation rules [39] is an interesting finding of this analysis which indicates the potentiality of OWL to support the MDD paradigm as a modeling language and as a transformation language as well.

Since ATL and QVT are the most predominant transformation languages, to analyze the evolution of their usage in the time could be likewise useful. Fig. 5 depicts that till 2015 (except in 2013), QVT (dotted curve) was preferred. Actually, in 2007 and 2011 the 100 % of the works used QVT. However, the graph in Fig. 5 that use percentages shows the trend to increase the usage of ATL in the most recent works.

We consider that is also useful to know what specific language of the QVT standard has been adopted. This analysis yielded that six works use the Operational Language [23], [33], [34], [40], [41], [42], but eight do not mention the specific language that they use. The lack of information hinders a more precise analysis, but the evidence indicates preliminarily that most works have used the Operational Language of the QVT standard.

RQ.4 -*What tools are used to transform CIM to PIM?*

As mentioned for the transformation languages, some papers point out that they carried out automatic transformations, but they do not provide any information regarding the transformation languages and tools. Taking into account this limitation, we found that there is not a wide variety of tools to support the model transformations. Eclipse has been used in 16 works. It means that 70 % of the 23 works that specify the tool applied to support the model transformation. The examination of the 16 works that adopt ATL as transformation language reveals that 13 of them adopted Eclipse and three of them do not specify the tool used. However, since that the ATL Plugin is the *de facto* tool to support the specification of ATL rules, we could assume that these three works use Eclipse as well.

Differently to ATL, 12 of 14 works that adopt QVT do not specify the tool adopted to support the model transformation. This fact could be considered as a gap of these works in terms of completeness. This type of information could be necessary to carry out a complete assessment of the approaches.

On the other hand, some plugins based on Eclipse have been developed, for example Papyrus Modeling Tool, Acceleo Plugin and MoDAr-WA Plugin. Furthermore, other tools have been developed, for example AMADEOS [4], Attribute Graph Grammar (AGG) tool [43] and BPSec-Tool [44].

RQ.5 -*How is the transformation carried out?*

The imprecise description of how the transformations are carried out is a recurrent gap of the analyzed approaches. Some papers point out that they carried out automatic transformations, but they do not provide any information regarding the transformation languages and tools adopted to support the model transformations. Thus, taking into account this gap, the analysis yielded that 16 works carried out manual transformations. However, we consider that regarding the transformation, the most interesting analysis is to explore how the way of carrying out model transformations has evolved over time. Fig. 6 depicts for each year the percentage of works with tool-support (automatic or semiautomatic) transformations (blue dotted curve) and manual transformations (red curve).

This examination reveals that except in 2014 and 2015, the number of works with tool-support transformations has been superior than the works with manual transformation. Besides, since 2016 there are more works that make use of tool-support transformations. This behavior could be motivated for the

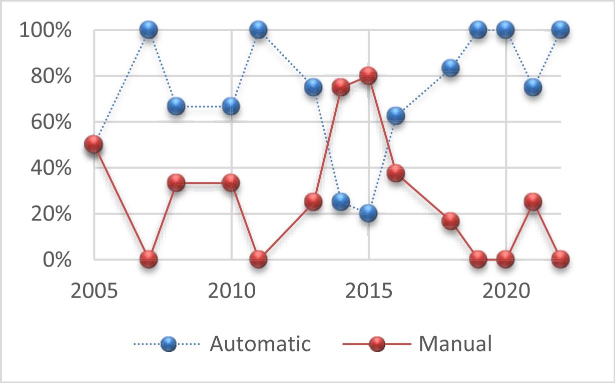


FIGURE 6. Comparison between manual and tool-support transformations.

TABLE 11. Most common transformation types.

Source	Target	Number
BPMN BPD	UML Class diagram	12
BPMN BPD	UML Use case diagram	12
BPMN BPD	UML State diagram	5
BPMN BPD	UML Sequence diagram	3
UML Activity diagram	UML Class diagram	6
UML Activity diagram	UML Use case diagram	5
UML Activity diagram	UML Sequence diagram	6
UML Activity diagram	UML Package diagram	3
UML Use case diagram	UML Class diagram	3
UML Use case diagram	UML Sequence diagram	5

maturity of the tools that support the model transformation. The experience of the researchers and practitioners with the technologies to support the model transformation might be another factor.

RQ.6 -*What transformations (Source-Target) have been carried out?*

This is another common question. We found 67 transformation types. The most common are the transformation *BPMN BPD to UML Activity diagram* and *BPMN BPD to UML Use case diagram.* The other common transformations are related to BPMN BPD and UML-based models. However, surprisingly *OWL Domain model -> UML Class diagram* is one of the most frequent transformations. Table 11 shows the most frequent transformation types and Table 12 shows the full list of transformations.

Focusing the analysis on the BPMN models, we found that they were transformed into 13 types of models and was the source model for one transformation *(OWL Business process model -> BPMN BPD*). This is indicative of the extensive adoption of BPMN models in the MDD context. Another interesting finding of our analysis is that 50 transformations include an UML-based model either as source or target model. It means that 75 % of the total transformations include an UML-based model. In addition, if we take into account the BPMN models, which are together with UML models a recommendation of OMG, we found that 56 transformations include one of the two types of models. It means 84 % of the total of transformations. These statistics confirm that the majority of works have focused their attention to adopt OMG recommended models in the MDD context. Furthermore, the statistics corroborate that BPMN and UML models are the standard for the CIM and PIM levels.

The main problem which remains in order to more effectively generate an IT system from business requirements is how to build a CIM so that it can be automatically transformed into a PIM [21]. In that sense, we realized that, in addition to the languages to represent the CIM and PIM levels, the definition of a specific strategy to construct the CIMs might be a key step to achieve the CIM to PIM transformation. We found different strategies, for example using Patterns and archetypes [21], defining guidelines to create the models [22], using a pattern-matching NLP technique [45], *Part Of speech* and *coreference resolution* to elaborate user stories [46], guiding the identification of business objects [47] and defining a set of Linguistic Syntax Patterns (LSP) and a business context to guide the business expert in the annotation of the business processes and alleviate the complexity of the identification of use cases [48].

RQ.7 -*What research types and research methods have been applied?*

We found only three *evaluation researches,* two of them conducted a controlled experiment with practitioners [4], [48] and the other applied an *industrial case study* [44]. This is strong evidence of the lack of empirical studies to demonstrate the impact of these approaches. Rodríguez *et al.* [44] assessed the opinion of the organization regarding the possibility of being able to express security in business process models; the learning curve associated with the application of the method and the use of the tools to facilitate the business analysts' work. Whereas Ben Ayed applies useful metrics to evaluate the transformations. In this work, the *Recall* is the ratio that indicates the capacity of a transformation to return all elements specified by the expert. The *Precision* is the ratio of real elements generated by the transformation that were identified by an expert. It indicates how accurate the transformation rules are in the generation of the target model. In addition, they used the *F-measure* to have the harmonic mean of recall and precision. These metrics have been applied by other approaches to evaluate the transformations [4].

Some papers [49] carry out a theoretical evaluation taking into account some criteria, such as: CIM coverage (business process, requirement and domain view), PIM coverage (Behavioral and structural), way of transformation (Automatic, manual, semi). Usually, they analyzed how other approaches fulfill these criteria and compare them with their own approach.

RQ.8 -*What quality attributes (productivity, efficacy, efficiency and cost) of the software development process are improved with the CIM to PIM transformations?*

The three approaches that conduct *evaluation researches,* did not assess the impact on the quality attributes of the

TABLE 12. Frequency of transformation types.

Source	Target	Frequency
BPMN BPD	UML Class diagram	12
BPMN BPD	UML Use case diagram	12
UML Activity diagram	UML Class diagram	6
UML Activity diagram	UML State diagram	6
BPMN-BPD	UML State diagram	5
UML Use case diagram	UML Sequence diagram	5
UML Activity diagram	UML Use case diagram	5
BPMN-BPD	UML Package diagram	3
UML Use case diagram	UML Class diagram	3
UML Activity diagram	UML Package diagram	3
OWL Domain model	UML Class diagram	3
BPMN-BPD	IFML diagrams	2
Data Flow Diagram (DFD)	UML Use case diagram	2
BPMN-CD	UML Use case diagram	2
BPMN-CD	UML Package diagram	2
UML-Class diagram	UML Package diagram	2
Emotional Model (Ecore metamodel)	UML Class diagram	1
UML Activity diagram	UML State machine	1
UML Activity diagram	Attribute Graph Grammar (AGG) (XML) - Task Description	1
E3value (Ecore Metamodel)	IFML (Ecore metamodel)	1
BPMN-BPD	Petri-Nets Business process model	1
Petri-Nets Business process model	UML (Activity Diagram) Service composition model	1
OWL Business process model	BPMN-BPD	1
OWL Business Process model	UML Use case diagram	1
OWL Business Process model	OWL Workflow model	1
OWL Description of rule-based ESs	UML Class diagram	1
User stories (Natural language)	UML Use case diagram	1
User stories (Natural language)	UML Class diagram	1
Marked content mapping model	ActivityModel, WorkflowModel & ActivityWorkflowMappingModel	1
DSL Simulated environment description language (SEDL)	UML Class diagram	1
i* framework Strategic dependency model (SD)	A-OOH Domain model	1
i* framework Strategic dependency model (SD)	A-OOH Navegational model	1
i* framework Strategic rationale model (SR)	A-OOH Domain model	1
i* framework Strategic rationale model (SR)	A-OOH Navegational model	1
Data Flow Diagram (DFD)	UML activity diagram	1
Data Flow Diagram (DFD)	UML Sequence diagram	1
Data Flow Diagram (DFD)	UML Domain model	1
Attribute Graph Grammar (AGG) (XML) - Task Description	Attribute Graph Grammar (AGG) (XML) - Navigation model	1
BPMN-BPD	SoaML model (three views)	1
BPSec: activity diagrams UML 2.0 profile	UML Class diagram	1
BPSec: activity diagrams UML 2.0 profile	UML Use case diagram	1
Feature model	Conceptual component model	1
Models of VIP Business Modeling Framework (BVM (Business Value Model), BIM (Business Information Model) and BPM (Business Process Model))	BPEL model	1
BMRL	UML Class diagram	1
BPMN-BPD	BMRL	1
UML Activity Diagram	BMRL	1
E3value	UML Use case diagram	1
BPMN-BPD	UML Sequence diagram	1
Business Rules (Natural language)	UML Class diagram	1
e3value Value model	UML Use case diagram	1
SoaML service architecture	UML Use case diagram	1
SoaML service choreography (UML Activity diagram)	UML Use case diagram	1
SoaML service architecture	UML package diagram	1
UML Package diagram	SOAML Software components model	1
UML Class diagram	SOAML Software components model	1
UML State diagram	SOAML Software Components model	1
UML State diagram	Web modeling diagram	1
UML Class diagram	Web modeling diagram	1
UML Use case diagram	Web modeling diagram	1
BPMN BPD	UML Activity diagram	1

TABLE 12. *(Continued.)* **Frequency of transformation types.**

UML Class diagram	UML Class diagram (MVC structured)	1
OWL Component model	UML Component model	1
OWL Business Process model	OWL Component model	1
BPMN-BPD	OWL Business process model	1
UML activity diagram	UML Sequence diagram	1
UML Use case diagram	UML Component model	1
Data Flow Diagram (DFD)	UML Package diagram	1

development process (*productivity, efficacy, efficiency and cost*).

This fact denotes the lack of empirical evidence about the positive impact of these approaches to enhance the software development process. This gap affects the extensive adoption of these approaches for the software industry. Hence, it is indispensable to conduct new studies which provide those empirical evidence to demonstrate the positive impact of the application of this type of approaches. Otherwise, the software industry will not be motivated to extensively adopt this type of approaches because it could be considered that MDD practices require an additional effort but without a clear benefit.

However, as a positive feature of the analyzed approaches, we found that they usually describe a case study to demonstrate their applicability. We found applications in a widespread range of domains, for example e-commerce, health processes, education and others application domain. That is a preliminary confirmation of the potential that these approaches have to improve the software development in terms of product quality and increase the productivity. However, as Sebastián *et al.* pointed out, it is necessary to demonstrate the advantages in terms of well-known metrics of using these approaches on these application domains with respect to traditional approaches [3].

RQ.9 -*What methodological aspects do these approaches deal with?*

We found very few works that include methodological aspects [44], [48]. Ben Ayed and Ben Abdallah explicitly define the roles and activities required to obtain requirement specifications (for example, by using UML Use case diagram) from business specifications (BPMN model) following the MDD practices. The definition of activities to specify and implement the transformation rules is a positive aspect of this proposal.

The approach defined by Ben Ayed and Ben Abdallah represents a prominent attempt to provide a methodological guide to adopt MDD-based proposals. However, we considered that some specific aspects could be also considered in order to achieve a more general methodological guide. For example, an activity is defined to validate the generated UML Use case diagram, however there is no activity to validate the input model (BPMN model) for this transformation. Therefore, the errors of the input model may affect the quality of the output model. Moreover, the adoption of only non-formal notations hinders the automatic validation and consistency checking of the models. We also consider that the task of implementing the transformation rules should be carried out by other roles instead of by software engineers. In order to implement transformation rules, it is required to have knowledge about transformation languages and tools to programming them. Since this knowledge is specific to the MDD domain, we consider that these MDD-specific tasks should be carried out by specialized roles or practitioners. The responsibility of this role could be not only to implement the transformation rules, but also to support the other roles with the tasks specific to the MDD approach, for example metamodeling, programming of transformation rules and models validation.

Rodriguez defines a set of stages, roles, tools and artifacts that help to create secure business process models and obtain specific artifacts for software development in an engineering and systematic approach [44]. Four stages are defined, the first three stages aim to guide the design of a business process, add security requirements to it and refine it. In the last stage, analysis classes and use cases are automatically generated. The correct implementation of these stages will contribute to make it easier to adopt this approach and to increase the probability of attaining a successful implementation. However, these activities aim to guide the adoption of this specific approach. Hence, some more-general activities are not included. For example, there are no activities related to metamodeling, definition of transformation rules and validation of models. As we aforementioned, this type of activities could be very useful for those development teams without experience with the MDD specifications.

Regarding roles, they define the roles of business analyst and security expert. These roles have specific responsibilities to create, refine and transform secure business process models. However, similarly to the case of the previous work we consider that it is advisable to define a role with the responsibility of implementing the transformation rules and to support the other roles with the tasks specific to the MDD approach, e.g., metamodeling, programming of transformation rules and models validation.

Regarding the artifacts, they only consider the models which are created, refined and generated. The explicit definition of the artifacts is a positive feature of this approach. However, other inputs and outputs of the process could be considered as artifacts as well. For example, metamodels, transformation rules, results of model validations and others. Thus, common practices to manage artifacts can be applied to them. For example, good practices to manage changes.

De Castro *et al.* define a set of stages and activities to facilitate the development of service-oriented applications [50]. This proposal also defines the roles with their respective responsibilities. However, as in the previous analyzed approaches, there are no roles with the responsibility to develop the MDD-specific activities.

Nonetheless, other works also include partial methodological specifications [39], [43], [51], thus it is evident that methodological dimension does not receive enough attention. The lack of methodological indications is a factor that could considerably affect the success of adopting these approaches. These specifications may avoid the arbitrary use of these approaches and contribute to applying them in a systematic and unambiguous manner [44]. This type of guidance could be particularly helpful for those software development teams which do not have experience (or do not know) with some specific practices of MDD. For example, activities like metamodeling, definition of transformation rules and execution of model transformations might be completely new for them. Therefore, a guide with some indications related to the fundamental concepts and practices may help to save a valuable time and to increase the probability of success when MDD is adopted.

V. CONCLUSION

This study put into focus that although the transformation from CIM to PIM does not receive the same attention as other transformations in the MDA context, the researchers have been struggling to bridge the gap between CIM and PIM. Evidence of their effort are the 65 transformation types that have been achieved. The application of a systematic mapping method made it possible to characterize how the transformation of CIM into PIM has been addressed in the past. Some of the most significant results are:

- BPMN models, UML activity diagrams and UML use case diagrams are the most adopted models at the CIM level. We found that the business view and the functional view are commonly covered, but few approaches deal with the domain view. Furthermore, few formal notations have been adopted in this level which may affect the automatic validation of the models and consequently their transformation.
- The fact that 44 works (83 % of total) adopt some UML-based is strong evidence that demonstrates the extensive use of UML to represent the PIM level. Similarly, as at the CIM level, few formal notations have been adopted to represent the models in this level.
- Regarding the transformation, the study yielded that the 84 % of the transformations include either a UML-based model or a BPMN model. This statistic confirms the extensive use of the OMG recommendations in the MDA context. Besides, the study confirmed that ATL and QVT are the most adopted transformation languages, 86 % of the works that adopted a transformation language adopted either ATL or QVT. In addition, we observed that since 2016 most works are applying tools to support the transformations. Finally, the study corroborated that Eclipse is the most adopted tool to support the model transformation.
- We found only three *evaluation researches.* This fact reveals the lack of empirical studies to demonstrate the impact of these approaches. The study yielded that *Precision, Recall* and *F-measure* are the metrics applied to evaluate the transformations.
- The study also revealed that the methodological dimension is rarely addressed by the analyzed approaches. The lack of methodological specifications may lead to the arbitrary use of these approaches and hinder their adoption in a systematic and unambiguous manner.

REFERENCES

[1] (2014). OMG. *Model Driven Architecture (MDA): MDA Guide rev. 2.0*. [Online]. Available: http://www.omg.org/cgi-bin/doc

[2] (2003). OMG. *MDA Guide Version 1.0.1*. [Online]. Available: http://www.omg.org/docs/omg/03-06-01.pdf

[3] G. Sebastián, J. A. Gallud, and R. Tesoriero, "Code generation using model driven architecture: A systematic mapping study," *J. Comput. Lang.*, vol. 56, Feb. 2020, Art. no. 100935.

[4] D. Brdjanin, S. Ilic, G. Banjac, D. Banjac, and S. Maric, "Automatic derivation of conceptual database models from differently serialized business process models," *Softw. Syst. Model.*, vol. 20, no. 1, pp. 89–115, Feb. 2021.

[5] G. Loniewski, E. Insfran, and S. Abrahão, "A systematic review of the use of requirements engineering techniques in model-driven development," in *Proc. Int. Conf. Model Driven Eng. Lang. Syst.* Springer, 2010, pp. 213–227.

[6] P. H. Nguyen, M. Kramer, J. Klein, and Y. Le Traon, "An extensive systematic review on the model-driven development of secure systems," *Inf. Softw. Technol.*, vol. 68, pp. 62–81, Dec. 2015.

[7] H. Tufail, F. Azam, M. W. Anwar, and I. Qasim, "Model-driven development of mobile applications: A systematic literature review," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 1165–1171.

[8] C. Raibulet, F. A. Fontana, and M. Zanoni, "Model-driven reverse engineering approaches: A systematic literature review," *IEEE Access*, vol. 5, pp. 14516–14542, 2017.

[9] S. Farshidi, S. Jansen, and S. Fortuin, "Model-driven development platform selection: Four industry case studies," *Softw. Syst. Model.*, vol. 20, no. 5, pp. 1525–1551, 2021.

[10] N. Silega, "Model-driven development of multi-agent systems: A systematic mapping," in *Proc. 22nd Ibero-Amer. Conf. Softw. Eng.*, 2019, pp. 307–320.

[11] H. R. Sharifi, M. Mohsenzadeh, and S. M. Hashemi, "CIM to PIM transformation: An analytical survey," *Int. J. Comput. Technol. Appl.*, vol. 3, no. 2, pp. 791–796, 2012.

[12] M. Habba, M. Fredj, and S. B. Chaouni, "Alignment between business requirement, business process, and software system: A systematic literature review," *J. Eng.*, vol. 2019, pp. 1–19, Oct. 2019.

[13] N. Kharmoum, K. E. Bouchti, N. Laaz, W. Rhalem, and Y. Rhazali, "Transformations' study between requirements models and business process models in MDA approach," *Proc. Comput. Sci.*, vol. 170, pp. 819–824, Jan. 2020.

[14] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. Electron. Workshops Comput.*, Jun. 2008, pp. 1–10.

[15] D. A. Meedeniya, "Correct model-to-model transformation for formal verification," Ph.D. dissertation, School Comput. Sci., Univ. St Andrews, St Andrews, U.K., 2013.

[16] D. Berardi, D. Calvanese, and G. De Giacomo, "Reasoning on UML class diagrams," *Artif. Intell.*, vol. 168, nos. 1–2, pp. 70–118, Oct. 2005.

[17] J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, Boeing, E. Kendall. (2006). *Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering*. W3C, E.P.T. (Feb. 11, 2006). [Online]. Available: http://www.w3.org/2001/sw/BestPractices/SE/ODA/

[18] (2014). OMG. *Ontology Definition Metamodel*. [Online]. Available: http://www.omg.org/spec/ODM/1.1/

[19] A. W. Brown, ''Model driven architecture: Principles and practice,'' *Softw. Syst. Model.*, vol. 3, no. 4, pp. 314–327, 2004.

[20] S. Beydeda, M. Book, and V. Gruhn, *Model-Driven Software Development*, vol. 15. Berlin, Germany: Springer-Verlag, 2005.

[21] S. Kherraf and W. Lefebvre Suryn, ''Transformation from CIM to PIM using patterns and archetypes,'' in *Proc. 19th Austral. Conf. Softw. Eng. (ASWEC)*, Mar. 2008, pp. 338–346.

[22] Y. Rhazali, Y. Hadi, and A. Mouloudi, ''Disciplined approach for transformation CIM to PIM in MDA,'' in *Proc. 3rd Int. Conf. Model-Driven Eng. Softw. Develop. (MODELSWARD)*, Angers, France, Feb. 2015, pp. 312–320.

[23] I. Essebaa and S. Chantit, ''Toward an automatic approach to get PIM level from CIM level using QVT rules,'' in *Proc. 11th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, Mohammedia, Morocco, Oct. 2016, pp. 1–6.

[24] S. Mujtaba, ''Software product line variability: A systematic mapping study,'' School Eng., Blekinge Inst. Technol., Karlskrona, Sweden, 2008. [Online]. Available: http://sites.google.com/site/drfeldt/mujtaba_apsec08_sysmap_spl_variabili.pdf

[25] K. Petersen, S. Vakkalanka, and L. Kuzniarz, ''Guidelines for conducting systematic mapping studies in software engineering: An update,'' *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.

[26] A. Noruzi, ''Google scholar: The new generation of citation indexes,'' *Libri*, vol. 55, no. 4, pp. 170–180, Dec. 2005.

[27] K. Gwet, ''Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity,'' *Stat. Methods Inter-Rater Rel. Assessment*, vol. 2, no. 1, p. 9, 2002.

[28] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical Methods for Rates and Proportions*. Hoboken, NJ, USA: Wiley, 2013.

[29] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, ''Requirements engineering paper classification and evaluation criteria: A proposal and a discussion,'' *Requir. Eng.*, vol. 11, no. 1, pp. 102–107, Mar. 2006.

[30] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, ''Systematic literature reviews in software engineering-a systematic literature review,'' *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009.

[31] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, ''Lessons from applying the systematic literature review process within the software engineering domain,'' *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007.

[32] Z. Graja, A. Regayeg, and A. H. Kacem, ''ForMAAD: Towards a model driven approach for agent based application design,'' in *Proc. Int. Workshop Agent-Oriented Softw. Eng.* Berlin, Germany: Springer, 2010, pp. 148–164.

[33] N. Laaz, ''Integrating domain ontologies in an MDA-based development process of e-health management systems at the CIM level,'' in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, 2020, pp. 213–223.

[34] N. Laaz, N. Kharmoum, and S. Mbarki, ''Combining domain ontologies and BPMN models at the CIM level to generate IFML models,'' *Proc. Comput. Sci.*, vol. 170, pp. 851–856, Jan. 2020.

[35] A. Y. Yurin, N. O. Dorodnykh, O. A. Nikolaychuk, and M. A. Grishenko, ''Designing rule-based expert systems with the aid of the model-driven development approach,'' *Expert Syst.*, vol. 35, no. 5, Oct. 2018, Art. no. e12291.

[36] K. Arrhioui, S. Mbarki, and M. Erramdani, ''Applying CIM-to-PIM model transformation for development of emotional intelligence tests platform,'' *Int. J. Online Eng.*, vol. 14, no. 8, p. 160, Aug. 2018.

[37] (2015). OMG. *Interaction Flow Modeling Language*. [Online]. Available: http://www.omg.org/spec/IFML/1.0/

[38] N. Kharmoum, S. Ziti, Y. Rhazali, and F. Omary, ''An automatic transformation method from the E3value model to IFML model: An MDA approach,'' *J. Comput. Sci.*, vol. 15, no. 6, pp. 800–813, Jun. 2019.

[39] Y. Che, G. Wang, M. Lv, and B. Y. Ren, ''Research on model transformation method oriented to model driven ERP system,'' *Adv. Mater. Res.*, vol. 670, pp. 208–215, Mar. 2013.

[40] I. Essebaa, S. Chantit, and M. Ramdani, ''MoDAr-WA: Tool support to automate an MDA approach for MVC web application,'' *Computers*, vol. 8, no. 4, p. 89, Dec. 2019.

[41] 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, pp. 55–81, 2014.

[42] 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. 9, p. 170, Sep. 2018.

[43] P. Valderas and J. V. Fons Pelechano, ''From web requirements to navigational design—A transformational approach,'' in *Proc. 5th Int. Conf. Web Eng.* Sydney, NSW, Australia: Springer, 2005, pp. 506–511.

[44] A. Rodríguez, E. Fernández-Medina, J. Trujillo, and M. Piattini, ''Secure business process model specification through a UML 2.0 activity diagram profile,'' *Decis. Support Syst.*, vol. 51, no. 3, pp. 446–465, Jun. 2011.

[45] M. Elallaoui, K. Nafil, and R. Touahni, ''Automatic transformation of user stories into UML use case diagrams using NLP techniques,'' *Proc. Comput. Sci.*, vol. 130, pp. 42–49, Jan. 2018.

[46] S. Nasiri, Y. Rhazali, M. Lahmer, and N. Chenfour, ''Towards a generation of class diagram from user stories in agile methods,'' *Proc. Comput. Sci.*, vol. 170, pp. 831–837, Jan. 2020.

[47] W. Khlif, N. Elleuch, E. Alotabi, and H. Ben-Abdallah, ''Designing BP-IS aligned models: An MDA-based transformation methodology,'' in *Proc. 13th Int. Conf. Eval. Novel Approaches to Softw. Eng.*, Funchal, Portugal, 2018, pp. 1–9.

[48] N. E. B. Ayed and H. B. Abdallah, ''From an annotated BPMN model to a use case diagram: DESTINY methodology,'' in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, 2020, pp. 379–390.

[49] B. Bousetta and O. T. El Beggar Gadi, ''A methodology for CIM modelling and its transformation to PIM,'' *J. Inf. Eng. Appl.*, vol. 3, no. 2, pp. 1–21, 2013.

[50] V. De Castro, E. Marcos, and J. M. Vara, ''Applying CIM-to-PIM model transformations for the service-oriented development of information systems,'' *Inf. Softw. Technol.*, vol. 53, no. 1, pp. 87–105, Jan. 2011.

[51] A. Harbouche, M. Erradi, and A. Mokhtari, ''Deriving multi-agent system behavior,'' *Int. J. Soft. Eng. Appl.*, vol. 7, no. 4, pp. 1–20, Jul. 2013.

NEMURY SILEGA received the Graduate and Ph.D. degrees from the University of Informatics Sciences, Habana, Cuba, in 2007 and 2014, respectively. Currently, he is the Research Leader of Southern Federal University, Taganrog, Russia. He has published more than 50 papers in international conferences and more than 20 journals. His current research interests include model-driven development, ontology-driven engineering, and business process modeling. In 2015, he was awarded with the Erasmus Mundus Scholarship for a postdoctoral stay with the University of Granada.

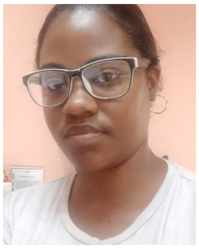
MANUEL NOGUERA received the Ph.D. degree from the University of Granada, Spain, in 2009. He is currently an Associate Professor in computer science with the University of Granada. He is one of the two principal investigators of the AVISAME (''NOTIFYME'') national research and development project, intended to foster healthy habits in elderly people. He was also the Leader of the CloudRehab Project, a mobile, cloud-based telerehabilitation platform for people with brain damage. Currently, he is also participating in several projects funded by the European Union, the national research agency of Spain and the Andalusian Regional Government. Most of his Ph.D. work background is on system and software architectures, groupware and ontology-driven engineering approaches to software development. He is also the Co-Founder of Everyware Technologies Ltd., a spin-off company specialized in the development of information systems and mobile computing solutions. He has also participated as a Management Committee Member of several COST Actions funded by the European Commission. He has published more than 70 papers in international conferences and more than 26 in well-reputed journals. He has also supervised five postdoctoral theses. His current research interests include the development of mobile monitoring platforms for health and sport domains.

YURI I. ROGOZOV received the B.S. degree in information and measurement technology from the Taganrog Radio Engineering Institute, Taganrog, Russia, in 1975, the Ph.D. degree, in 1984, and the Doctor of Technical Sciences degree, in 1991.

Since 1975, he has been working as an Engineer, an Assistant, a Senior Lecturer, and an Associate Professor. Since 1991, he has been as an University Professor. In 1992, he was elected to the position of the Dean of the Faculty of Automation and Computer Engineering. From 1994 to 1999, he worked as the Vice-Rector for Informatization. Since 1999, he has been the Head of the Department of System Analysis and Telecommunications, Taganrog State Radio Engineering University. He has published more than 140 scientific papers, including a monograph and received more than 70 copyright certificates for inventions. His research interests include methodology of system analysis, methods of logical and circuit design, integrated circuits in the logical-arithmetic basis, and new information technologies. He is a well-known specialist in the field of electronics and microcircuitry, who proposed a new principle for the design of integrated circuits, developed methods for the logical and circuit design of digital devices of large integrated circuits. He is also a Honorary Worker of Higher Education, in 2002.

VYACHESLAV S. LAPSHIN received the B.S. degree in information and measurement technology from the Institute of Computer Technologies and Information Security, Southern Federal University, Taganrog, Russia, in 2017, where he is currently pursuing the Ph.D. degree.

He is also working as an Assistant Professor at the Department of System Analysis and Telecommunications, Institute of Computer Technologies and Information Security, Southern Federal University. He has published more than 40 scientific papers and has received four copyright certificates for registration of intellectual property. His research interests include issues related to the creation of complex technical systems, in particular, the development of projects for intelligent transport systems and their subsystems, issues related to the development of functioning algorithms for adaptive systems, processes of building self-organizing systems, and the development of methods for designing complex technical systems.



TAHUMARA GONZÁLEZ received the Graduate degree from the University of Informatics Sciences, Habana, Cuba, in 2012, where she is currently pursuing the master's degree. She has a wide experience working in software development projects. Her current research interests include the fields related to model-driven development, advanced programming, and ontology engineering.