Analyzing CIM to PIM Transformations Using the WRSPM model

FÁBIO LEVY SIQUEIRA[1]
PAULO SÉRGIO MUNIZ SILVA[2]

Escola Politécnica da Universidade de São Paulo
Departmento de Engenharia de Computação e Sistemas Digitais
Av. Prof. Luciano Gualberto, trav.3, no 158 - 05508-900 - São Paulo - SP - Brazil
[1]levy.siqueira@gmail.com
[2]paulo.muniz@poli.usp.br

Abstract. Model-Driven Architecture (MDA) is a framework for software development focusing on models and model transformation. One of its models is the computation independent model (CIM), which describes the system environment and requirements. Representing this model and transforming it into a platform independent model (PIM) typically involves Requirements Engineering approaches and diagrams. Although there are some approaches that propose CIM to PIM transformations, it is not clear how Requirements Engineering concepts are related to MDA models. To improve the understanding of MDA and to define which artifacts are expected from a Requirements Engineering perspective in CIM to PIM transformations, this paper analyzes some proposals, found in a survey, and proposes a mapping between a requirements reference model, the WRSPM model, and Model-Driven Architecture models.

Keywords: computation independent model, platform independent model, MDA, requirement, specification, WRSPM model.

(Received January 30th, 2012 / Accepted March 26th, 2012)

1 Introduction

Model-Driven Architecture (MDA) is a framework for software development that specifies three models [21]: a computation independent model (CIM), a platform independent model (PIM), and a platform specific model (PSM). Because it emphasizes the platform concept, the MDA guide briefly describes the CIM and its transformation to PIM. Nevertheless, there are several different proposals for CIM to PIM transformations, for example, [1], [2], [5], [10], [14], [20], [25], [27], [28], [31], [32], [34], [36], [37], and [38].

Because the requirements for the system are modeled in the computation independent model [21], several Requirements Engineering (RE) approaches and diagrams are used to represent this model and to transform it into PIM. However, RE approaches and diagrams can be also used in the platform independent model. For example, in [5] and [27], a use case diagram is one of the components of the proposed PIM. Because MDA specifies few characteristics and requirements for its models, it is not clear how RE concepts are related to MDA models. Consequently, it can be difficult for an MDA user to define which RE artifact should be used for which MDA model, or to extend an existing CIM to PIM transformation. Moreover, encountering the same model being used by different approaches at the same level of detail could be considered a contradiction. However, there are studies that consider, for example, use cases as part of the CIM [8, 14] or as part of the PIM [5, 27].

In order to define which types of artifacts are expected from a RE perspective from each MDA viewpoint, this study analyzes the models used by some CIM to PIM transformations, found in a survey, considering a requirements reference model, the WRSPM model (meaning World, Requirement, Specification, Platform, and Machine) [9]. This model was selected because it

is method independent and it proposes a set of software development artifacts from a RE perspective. Based on this analysis and on the definitions presented by the MDA guide and the WRSPM model, it is proposed a mapping between WRSPM model artifacts and MDA models. In addition to helping MDA users apply the framework, this also improves the understanding of MDA from an RE perspective[1].

This paper is organized as follows: in Section 2, the MDA and its models are discussed, and in Section 3, the WRSPM model is presented. The survey and the analysis of CIM to PIM transformations are presented in Section 4. In Section 5, it is proposed a mapping between the WRSPM model and MDA models. Next, in Section 6 it is presented the related works. Finally, the conclusions are presented in Section 7.

2 Model-Driven Architecture (MDA)

Model-Driven Architecture (MDA) is a software development approach that focuses on models and model transformation. Its primary goal is to improve portability, interoperability, and reusability through the architectural separation of concerns [21]. In order to accomplish this goal, MDA specifies three models. Figure 1 represents these models and their transformations. The computation independent model (CIM) is the first model to be created in an MDA approach. It represents the system requirements by describing the situation in which the system will be used [21]. However, it describes the environment in which the system will operate, rather than the system structure. As a result, the CIM does not specify technological or implementation details and it can even be implemented without software systems [17].

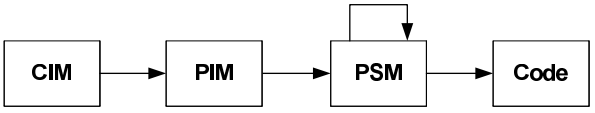

Figure 1: Models (boxes) and transformations (arrows) in MDA.

MDA briefly describes the CIM and its transformation into PIM. The only requirement is that the PIM should be traceable to the CIM [21]. Nevertheless, the PIM is obtained from the CIM by taking into account system details. Although it is more concrete than the CIM, its main characteristic is its independence from a specific platform. The platform concept is essential to MDA and it is the difference between the PIM and the PSM. According to Miller and Mukerji, a platform is "a set of subsystems and technologies that provide a coherent set of functionality through interfaces and specified usage patterns" [21, p. 2-3]. Therefore, a platform can represent operational systems, programming languages, or even development practices [17]. However, the concept of platform independence is a quality that can have several degrees [21]. A PIM can exhibit a complete independence to platforms, while another PIM can be dependent to a specific platform (for instance, web services or CORBA), but still exhibit independence to other platforms (for instance, Microsoft .NET or Enterprise Java Beans).

The last model defined by MDA is the PSM. It is obtained from the PIM by taking into account the details of a specific platform. However, other platform decisions may be necessary to transform this model into executable code. In other words, the PSM can be either an implementation model, which can be transformed into code, or a PIM, and then transformed into another PSM by taking into account a different platform.

3 The WRSPM Model

While the MDA focuses on software development models considering a specific approach, the WRSPM model (meaning World, Requirement, Specification, Platform, and Machine) provides a common basis for discussing software development artifacts. It is based on the ideas and concepts presented by Jackson and Zave [11, 12, 35], and emphasizes the difference between requirements and specifications - thus having a Requirements Engineering perspective. Artifacts are classified into five groups according to their importance to the environment - the part of the world with which the system will interact, and where the effects of the system will be observed and evaluated [12] - or to the system itself, as represented in Figure 2. These artifacts are [9]: domain knowledge, containing known and presumed information about the environment; requirement, indicating what the client needs from the system, described as effects on the environment; specification, enabling the system that satisfies the requirements to be built; program, implementing the specification; and platform, enabling the system to be programmed.

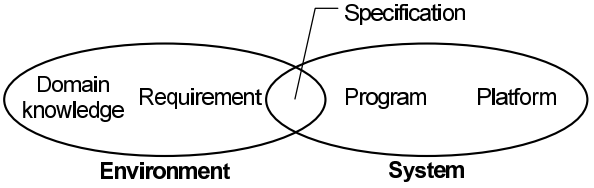

Figure 2: The WRSPM Model [9].

Each of these artifacts can be viewed as a descrip-

[1]This paper is an extended version of [30].

tion that uses a specific vocabulary comprising of primitive terms. Some of these terms are shared among artifacts, just as the phenomena that they aim to denote are also shared. These phenomena represent states, events, and individuals that are organized into four classes by the model: the ones that belong to the environment and are visible or invisible to the system, and, similarly, the phenomena that belong to the system and are visible or invisible to the environment. According to this division, only the visible phenomena are shared between the environment and the system. For example, in a library computer system, an environmental phenomenon that is invisible to the system is the arrival of new books; a visible environmental phenomenon is the registration of these new books; a system phenomenon visible to the environment is the availability of these books for lending; and, finally, an invisible system phenomenon is any change to the database record regarding these books.

Given these classes, the domain knowledge artifact restricts the environmental phenomena. In addition, it can also restrict the relationship between environmental phenomena and phenomena belonging to the system that are visible to the environment [9]; for instance, in the example of the library system, a restriction is that a book becomes unavailable when it is lent. The requirement artifact adds other restrictions to the ones defined by the domain knowledge artifact, considering stakeholder needs. In other words, the requirements are decided by the stakeholders. Using the library example once more, a restriction such as a library member only being allowed to borrow three books at any given time can be part of the requirement artifact. On the other hand, the platform artifact restricts the phenomena that can be executed by the system, while the program artifact describes a class of possible system phenomena with regards to the program. In the middle of the environment and the system, there is the specification artifact that uses their common vocabulary to determine what the system must accomplish, with regards to the environment.

4 CIM to PIM Transformations

MDA can be applied using different processes and methods [16], as it does not specify or impose a software development process. The framework only specifies few characteristics and requirements for its models, and discusses model transformation. Therefore, each MDA approach can organize the three MDA models differently into software development artifacts. An artifact is defined here as "the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system" [24, p. 203][2]. By this definition, it is possible for an artifact to have more than one model, or for a model to be in more than one artifact.

Analyzing the MDA models considering the WRSPM model, both CIM and PIM can be related to Requirements Engineering artifacts. The CIM represents the environment in which the system will operate, thus not describing system details. Depending on the platform, the PIM may represent the system to be built, without describing implementation details (the implementation details would be described considering a specific platform, for instance).

Although some approaches use a CIM as a requirement model and a PIM as a specification model, the CIM and the PIM are not equivalent to these artifacts. Depending on the approach, different mappings are possible. Moreover, each MDA model can be broken down into several models, and these models may be positioned in different WRSPM artifacts. For instance, the domain knowledge artifact may contain an as-is process model (representing the existing process); the requirements may contain a goal model [33]; and the specification may contain a use case model. If an MDA approach proposes a CIM consisting of an as-is process model, a goal model, and a use case model, it would have elements from all these three artifacts. The mapping is even more complex because some sub-models (for example, goal model, use case model, or process model) can be used by different artifacts. For instance, a use case model may be in a domain knowledge artifact as a business use case, in a requirement artifact as a use case at a summary level [3], in a specification artifact as a use case at a goal level [3], or even in the program artifact as a use case at a subfunction level [3]. Therefore, each MDA approach may use a metamodel differently, either in the dimension of MDA models, or in the dimension of WRSPM artifacts.

4.1 Survey

To analyze the relationship between CIM and PIM models, and WRSPM artifacts, we conducted a survey of CIM to PIM transformations considering some important article databases: ACM, Elsevier, IEEE, and Springer[3]. We carried out a search for papers that had the words "CIM" or "Computation Independent", and "PIM" or "Platform Independent" in their titles or ab-

[2] A definition of an artifact from the WRSPM model was not used, because the model only states that an artifact can be viewed "primarily as a descriptions written in various languages" [9, p. 38] - and a model is more than a description [11]. Still, the definition used seems to be compatible with the WRSPM model.

[3] This survey was conducted in April 24, 2012.

stracts. Only papers that propose a CIM to PIM transformation, describing both models, were analyzed. In addition, when there was more than one paper written by a research group, only the most recent study or the study that described the CIM to PIM transformation was considered. The results are presented in Table 1, indicating the number of papers analyzed and ignored. Considering the 48 results obtained in the survey, 7 papers were ignored since they are from different knowledge areas ("Other area") - biology, business, manufacturing, materials, metallurgy, and networks -, thus the searched words have a different meaning; 2 results are repeated papers ("Repeated"); and 7 describe previous versions of a proposal ("Previous work"). In addition, 17 papers do not propose a CIM to PIM transformation ("No transf."). This is due to: a) there is no description of the metamodels and the transformation (1 paper); b) there is a proposition of another type of transformation, using other models (2 papers); c) there is only the use of an existing transformation (1 paper, which applied the approach in [5]); and d) there is no proposition of a transformation - for instance, they discuss the meaning of the CIM [15], propose frameworks (e.g., [7] and [4]), or describe a project developed using an MDA approach (e.g., [18]) - (13 papers). Therefore, the survey obtained 15 papers that propose a CIM to PIM transformation.

Table 1: Results of the survey.

Results		*IEEE*	*ACM*	*Springer*	*Elsevier*
Ignored	Repeated	-	2	-	-
	Previous work	2	-	4	1
	No transf.	4	2	9	2
	Other area	2	-	4	1
Analyzed		7	-	6	2

Most of the CIM to PIM transformations (9 of them) focus on service-oriented systems, obtaining representations of services as PIM. Touzi, Bénaben, and Pingaud [31] propose a transformation from a business process, used as CIM, into a service oriented architectural model, used as PIM. The authors propose a business process metamodel and some transformation rules, obtaining a PIM, which is composed by three views: a services view, an information view, and a process view. The services view describes the services that will be used; the information view represents the data exchanged between the services; and the process view represents the interaction and coordination between services.

Using more than one model as CIM, Khreaff, Lefebvre, and Suryn [14] propose using two activity diagrams for this model: a business process model and a requirements model. The business process model represents the activities that must be executed independently of their automation, while the requirements model represents use cases, describing the activities considering the system as an actor. The PIM is obtained from the requirement model, and it is composed by a component diagram and a stereotyped class diagram. Also using two models as CIM, Che, Wang, and Ren [2] propose a software development approach based on MDA. The CIM is composed by a model representing the business, the CIM-Business, and a model capturing the requirements from the business, the CIM-ESA. Each model has three sub-models: a function process model, an information model, and an organization model. The PIM has a business view, representing business objects, and data and information models.

Using three models as CIM, Weber proposes technology independent CIM and PIM for message-based communication of information systems [34]. The CIM considers a memo model (MM), which is a data model representing immutable documents (called memos), a memo-order model (MOM), representing memos and time relations between them, and a memo-flow model (MFM), representing memos and actions that generate memos. The memos are considered as messages in the PIM, resulting in a data type interchange model, which is obtained from the CIM. In [10] it is not presented a full transformation, but a pattern that maps one type, used by a model in the CIM, to a state machine, which is part of the PIM. The proposed CIM is composed by three models that are represented using a UML profile. The business domain view (BDV) represents the existing business processes as use cases. This model is used to create the business requirements view (BRV), representing the collaboration between the business entities in two types of use cases. Finally, the business transaction view (BTV) defines "a global choreography of information exchanges and the document structure of these exchanges" [10, p. 58], represented by an activity diagram. The BTV uses some business transaction types, and, in the proposal, one of these types is mapped to two state machines, that serve as patterns. Each state machine is considered as part of the PIM, specifying the transaction for each entity involved in the transaction.

Differently from these studies, in [1] it is proposed using service features both as a requirements representation and as a design element. These service features are represented in a graph that describes two types of relationships: constraints and refinements. The high-level service features are part of the CIM, while the low-level service features are part of the PIM. However, it is not

specified what is the necessary refinement level to obtain a PIM.

Some studies also focus on service-oriented systems, but they consider business value models as part of the CIM. Zdravkovic and Ilayperuma [36] propose an approach to specify and design business services, using business value models as CIM. Therefore, the CIM is composed by three models: the value chain specification (VCS), describing the value-adding processes; the business process specification (BPS), describing economical events, resources, and agents in each process; and the business event specification (BES), representing the business services. On the other hand, the PIM is composed by two models: a class diagram, using a UML profile, and an activity diagram. The approach defines a mapping to obtain the PIM models from the CIM. Similarly, Zhang and Jiang [37] propose a CIM to PIM transformation to obtain a BPEL model, which is considered as a PIM. The CIM is composed by two models: a business value model (BVM), and a model that has an information systems perspective, composed by a Business Information Model (BIM) and a Business Process Model (BPM). The Business information Model traces concepts of the BVM to information systems, while the Business Process Model both integrates the other diagrams and represents the sequence of activities, through different viewpoints. Finally, [5] describes a CIM to PIM transformation that is part of a method to develop service-oriented systems. The CIM is a business model, composed by two models: a value model, and a business process model, using BPMN. On the other hand, the PIM is an information systems model, composed by 4 models: a use case model, adapted to represent the business services to be implemented; an extended use case model, detailing the business services; a service process model, representing the activities carried out by a service; and a service composition model, representing the relationship between services and other entities. The study discusses some transformation rules, using ATL (ATLAS Transformation Language [13]), and proposes a tool.

Some approaches consider other domains, instead of service-oriented systems. Valderas, Fons, and Pelechano [32] propose an approach to develop web applications using MDA. The CIM is represented by two sub-models: a task taxonomy model and an activity model. In the first sub-model, the tasks executed by users must be refined until they are executed by the system or the user - something they call elementary tasks. Then the elementary tasks are described in a UML activity diagram and in information templates that describe the information exchanged between the user and the system. From these two models, it is proposed a transformation to obtain a navigation model (three other models are not detailed). This model represents the views of a class diagram, called context, and the links between them, representing how one context can be obtained from another. Considering the domain of business process execution, in [28] it is proposed an approach to model and specify flexible business processes, using process versioning. At CIM level, a metamodel represents five views: process, functional, operational, organizational, and informational. Each view has a main meta-class that allows creating different versions (or alternatives). A specific version of the model is then transformed into a PIM, which is composed by two models: a business process model represented using an extended BPMN metamodel to handle versioning, and a Petri net with objects - which is not discussed in the proposal - to formalize, validate and simulate the business process. Prat, Akoka, and Comyn-Wattiau [25] propose a CIM to PIM transformation in the domain of knowledge engineering. The CIM used is the knowledge model defined by the development method CommonKADS (even though the method uses other models). This model "provides an implementation-independent description of knowledge involved in a task" [25, p. 3], and it is transformed into a PIM composed by a Production Rule Representation (PRR) model, defined by OMG [23], and by class and activity diagrams - which are not described.

Differently from these approaches, three studies do not consider specific domains. In [27], the CIM is composed of a business process model, using the UML activity diagram. This model is transformed, using QVT (Query/View/Transformation [22]), into two models that are part of the PIM: a use case diagram and a class diagram. The use cases must be detailed to obtain actions, and the class diagram is considered as an initial analysis model. This approach can be also used to deal with security requirements, as described in [26]. The CIM can use a specific UML activity diagram profile representing security requirements, and the transformation results in models considering this non-functional requirement. In another approach, [20] presents a transformation to obtain a UML class diagram, which is considered as a PIM. Two diagrams compose the CIM: a business process model and a concepts model (representing domain concepts and their attributes). These two models are combined into an intermediate model, which is transformed into a UML communication diagram. This diagram is then transformed automatically into a class diagram. Finally, Zhang et al. [38] propose a transformation from a feature model, used as a

Table 2: Mapping of MDA models into WRSPM artifacts in the studies analyzed.

Study	*DomainKnowledge*	*Requirement*	*Specification*	*Program*	*Platform*
[1]	-	CIM	PIM [a]	PIM [a]	PIM [a]
[2]	CIM (CIM-Business)	CIM (CIM-Business)	CIM (CIM-ESA)	PIM	-
[5]	CIM	CIM	PIM (UC and Extended UC)	PIM (Service process model and Service composition model)	-
[10]	CIM (BDV)	CIM (BRV)	CIM (BTV)	PIM	-
[14]	CIM (Business process model)	-	CIM (Requirements model)	PIM	-
[20]	CIM	-	CIM	PIM	-
[25]	CIM	CIM	CIM	PIM	-
[27]	CIM	CIM	PIM (Use case)	PIM (Class diagram)	-
[28]	CIM	CIM	PIM	-	-
[31]	CIM	CIM	CIM	PIM	-
[32]	CIM (Taxonomy model)	CIM (Taxonomy model)	CIM (Activity model)	PIM	-
[34]	CIM (MM and MOM)	CIM (MM and MOM)	CIM (MFM)	PIM	-
[36]	CIM (VCS and BPS)	CIM (VCS and BPS)	CIM (BES)	PIM	-
[37]	CIM (BVM)	CIM (BIM)	CIM (BPM)	PIM	PIM
[38]	-	CIM	CIM	PIM	-

[a] Depends on the details of the features.

CIM, to a component model, used as a PIM. The feature model represents relationships between features, i.e., a set of requirements. To obtain the PIM, the authors propose an approach that relates features to responsibilities, which represents a set of program specifications, and the responsibilities to components. Therefore, this approach uses an intermediate model that is neither CIM nor PIM.

4.2 Analysis

Table 2 presents relationship between the MDA models and the WRSPM artifacts for the 15 approaches found in the survey. The CIM and PIM sub-models are only described when they are spread into several artifacts, thus representing which sub-model is in which artifact.

In general, the CIM comprises the domain knowledge, requirement, and specification artifacts, while the PIM is in the program artifact. However, some approaches have a different mapping. In [1] and [38], the CIM is not in the domain knowledge artifact. In both proposals, the CIM represents features, describing what the stakeholders want for the system. Therefore, these models do not describe presumed information about the environment - they only describe requirements. Moreover, in [1] a single feature model is used as both CIM and PIM. The difference between the service features (represented in these models) is its refinement, although it is not specified what is the criterion to be in each model. Consequently, in this proposal the PIM can be also in the specification model - depending on the criteria. Moreover, the PIM can contain platform information depending on the feature obtained through an operationalization refinement. This category of refinement details a feature considering the target system, defining operations, data representations, structures, constraints, and/or agents [1]. Therefore, depending on the solution considered, it may contain platform information. Nevertheless, this model may be considered as a PIM, as "platform independence is a matter of degree" [21, p. 2-5]. Other platforms may be considered in the PSM, for instance, a vendor specific platform. Another study in which the PIM considers platform information is [37]. Because the PIM is described in BPEL, it contains information about a specific platform: web services.[4]

Similarly to [1], in [5], [27], and [28], the PIM is considered as part of the specification artifact. In [28], this is due to the fact the PIM only represents a version of the business process, therefore it does not describe a

[4] Although web services can be considered as a design strategy, they were considered here as a platform based on [29] and on the definition of platform provided by MDA. On the other hand, service orientation, object orientation, and the use of production rules, as solution strategies, were not considered as platforms.

program. In [5] and [27] the PIM is part of the specification artifact because it is composed by a use case diagram, describing the interaction between system and actors. At one hand, these models can be mapped to the specification artifact, as they represent use cases at goal level. On the other hand, it would be possible to map these models to the requirement artifact, as a use case diagram does not describe the behavior of its use cases. Nevertheless, they were both mapped to specifications: in [27], because the use case diagram is part of a use case model that must contain other diagrams to represent the details of user and system interaction; in [5], because this model is extended to represent services - and not just use cases.

Another study that has use cases as one of its models is [14]. However, differently from [5] and [27], a use case is part of the CIM. Therefore, different studies use a model as either CIM or PIM. Nevertheless, this is not a contradiction. Because the specification is at the interface between the environment and the system, this artifact has both an environment description and a system description [11]. In addition, this artifact is free of technological details, or at least it should be [35]. Therefore, a specification defines a model of the system independent of any platform. Thus, the content of a specification can be seen as a CIM or the most basic PIM - considering that a software development project using MDA may use several platforms, and, consequently, several PIMs.

Finally, in [14] and in [20] there is no model in the requirement artifact. In both approaches, because the business process model represents the processes without the system, it is in the domain knowledge artifact. In [20], it is considered that all processes should be automated. Therefore, this model and the conceptual model are also used as a specification [5]. On the other hand, in [14] there is another model, an activity diagram, which is in the specification artifact since it specifies the behavior of a use case. Therefore, in this approach the analyst must be able to create the specification model directly from the domain knowledge.

[5] Similarly to [20], in [27] the business process model does not describe the system, and the approach considers that all processes should be automated. However, in [27] the business process is not sufficiently detailed to be considered as a specification (the approach generates a use case model, which can be considered in the specification artifact). Therefore, it was considered in both the domain knowledge and requirement artifact. On the other hand, in [20], the business process model is more detailed, and there is a concepts model that provides additional information. Therefore, these two models can be also considered as a specification.

5 Mapping MDA Models onto WRSPM Artifacts

Table 3 presents the mapping of MDA models onto software development artifacts, considering the WRSPM model. This mapping was based on the generalization of the previous analysis of existing CIM to PIM transformations, and also on the definitions presented by the MDA guide and the WRSPM model.

Table 3: Mapping of MDA Models onto WRSPM Artifacts.

		MDA		
		CIM	PIM	PSM
WRSPM	Domain Knowledge	X [a]		
	Requirement	X		
	Specification	X	X	
	Program		X [b]	X [b]
	Platform		X [a, b]	X [a, b]

[a] This model cannot be in this artifact alone.
[b] The artifact cannot have only this model.

All the artifacts within the environment boundary are related to the computation independent model, as MDA defines the CIM as the model that represents the environment. Therefore, a CIM can be in the domain knowledge, the requirement, or the specification artifacts, but not necessarily in all of these artifacts. MDA specifies that this model "may hide much or all information about the use of automated data processing systems" [21, p. 3-1]. Consequently, a CIM may not be in a specification if all the information about the use of the system is hidden - what is the case in [5] and [27]. On the other hand, it cannot be in the domain knowledge alone, as this artifact does not have the concept of a system that exists in a CIM.

As previously discussed, the PIM may also be in a specification artifact when it is free of technological details (for instance, in [1], [5], and [27]). Nonetheless, if a PIM contains technological details, it will be in the program artifact - what is the case for almost all the transformations analyzed. Similarly, when a PSM is considered a PIM (in a transformation into another PSM), it is necessarily part of the program, as it already takes into account phenomena that are only visible to the system. In this situation, the PIM also have information about the platform used by the previous transformation. However, a PIM can be linked to the platform even if it is not a PSM: some platform information may be assumed and used directly by a PIM (for instance, in [37]). As discussed in Section 2, the MDA concept of platform independence does not necessarily mean that the model is independent of all platforms. Therefore, the PIM can be also part of the platform artifact, even though a PIM can never possess all the platform infor-

mation, nor can it be this artifact alone. At least the implementation model - the PSM that can be transformed into code - will also be in the platform and in the program artifacts. Thus, the PIM also cannot have all the content of the program artifact.

Finally, because the code is also part of the program, the PSM cannot be the sole content of this artifact. The PSM is also in the platform artifact, which can be seen as all the platforms that are used by the system. Some of these platforms are used to transform a PIM into a PSM; others may be assumed in the PIM. Although the PSM has some platform information, it cannot be the platform alone; it must also be in the program to be a model of the system. In addition, it cannot be the entire platform as there is also part of it in the code.

6 Related Work

Another study that proposes mapping involving MDA models is presented in [8]. That study considers the metamodels of OMG standards related to MDA, mapping them onto the Zachman framework, an enterprise architecture framework. Although it does not consider RE artifacts, it discusses the relationship between several diagrams defined by the UML metamodel that are used in RE artifacts. While the activity diagram is mapped onto all MDA models, the use case diagram is only mapped onto the CIM. The idea is that the use case diagram cannot be used as a specification because it does not represent the details of a system and user interaction. In other words, the use case diagram represents information about the domain knowledge or the requirement artifact. The specification would be represented in other diagrams, for instance, an activity diagram, which can represent the behavior of a use case [24].

Regarding to the proposed mapping for the CIM, in [15] it is presented another interpretation of its content, from an Information System perspective. The authors argue that the CIM consists of two abstractions: An abstraction of human information processing (HIM), representing the business knowledge, and an abstraction of software information processing (AIM), representing a software (or computational) interpretation of the business knowledge (as rules, patterns, and algorithms). Considering the WRSPM model, the HIM abstraction could be represented in the domain knowledge and/or the requirement artifacts, while the AIM abstraction could represented in a specification artifact. Therefore, the proposed mapping is compatible to the definition presented in [15].

7 Conclusion

This study analyzes existing CIM to PIM approaches found in a survey, mapping the proposed MDA models onto WRSPM artifacts. Based on this mapping and on the definitions provided by MDA and the WRSPM model, this study also proposes a general mapping of MDA models onto WRSPM artifacts. The relationship between these models and artifacts is not one-to-one: an MDA model may be in more than one WRSPM artifact, and an artifact may have more than one model. Therefore, each MDA approach may have models relating differently to the WRSPM artifacts.

This mapping contributes to the understanding of existing MDA approaches from a Requirements Engineering perspective. Although these approaches propose different models, in general, the CIM comprises the domain knowledge, requirement, and specification artifacts, while the PIM is in the program artifact. Moreover, this mapping highlights the possible content of the CIM and its distinction from the PIM, based on the difference between requirements and specifications. Consequently, it may also help future MDA approaches in defining the content for each MDA model. For instance, it is possible to use existing requirements to specification transformations, e.g., [6] and [19], as a starting point for an MDA approach.

Acknowledgments

This work was partly supported by *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP), grant no. 2007/58489-4.

References

[1] Cao, X., Miao, H., and Xu, Q. Modeling and Refining the Service-Oriented Requirement. In *Proceedings of the 2008 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, pages 159–165. IEEE Computer Society, 2008.

[2] Che, Y., Wang, G., and Ren, B. Research on Application of Model-Driven Architecture in the Development Process of Enterprise Information System. In *4th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, 2008.

[3] Cockburn, A. *Writing Effective Use Cases*. Addison-Wesley, 2000.

[4] Cortellessa, V., Di Marco, A., and Inverardi, P. Integrating Performance and Reliability Analysis in

a Non-Functional MDA Framework. In *Fundamental Approaches to Software Engineering*, volume 4422 of *Lecture Notes in Computer Science*, pages 57–71. Springer Berlin / Heidelberg, 2007.

[5] De Castro, V., Marcos, E., and Vara, J. M. Applying CIM-to-PIM Model Transformations for the Service-Oriented Development of Information Systems. *Information and Software Technology*, 53(1):87 – 105, 2011.

[6] de la Vara, J., Sánchez, J., and Pastor, O. Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems. In *Advanced Information Systems Engineering*, volume 5074 of *Lecture Notes in Computer Science*, pages 213–227. Springer Berlin/Heidelberg, 2008.

[7] Domínguez-Mayo, F. J., Escalona, M., and Mejías, M. Quality Issues on Model-Driven Web Engineering Methodologies. In *Information Systems Development*, pages 295–306. Springer New York, 2011.

[8] Frankel, D., Harmor, P., Mukerji, J., Odell, J., Owen, M., Rivitt, P., Rosen, M., and Soley, R. The Zachman Framework and the OMG's Model Driven Architecture. *Business Process Trends*, September 2003.

[9] Gunter, C., Gunter, E., Jackson, M., and Zave, P. A reference model for requirements and specifications. *Software, IEEE*, 17(3):37–43, may/jun 2000.

[10] Huemer, C. and Zapletal, M. A State Machine executing UMM Business Transactions. In *Proceedings of the 2007 IEEE International Conference on Digital Ecosystems and Technologies*, Cairns (Australia), 2007. IEEE Computer Society, IEEE Computer Society.

[11] Jackson, M. *Requirements & Specifications: a Lexicon of Practice, Principles, and Prejudices*. ACM Press / Addison-Wesley, New York, NY, USA, 1995.

[12] Jackson, M. The Meaning of Requirements. *Annals of Software Engineering*, 3:5–21, 1997.

[13] Jouault, F. and Kurtev, I. Transforming models with atl. In Bruel, J.-M., editor, *Satellite Events at the MoDELS 2005 Conference*, volume 3844 of *Lecture Notes in Computer Science*, pages 128–138. Springer Berlin / Heidelberg, 2006.

[14] Kherraf, S., Lefebvre, E., and Suryn, W. Transformation from CIM to PIM Using Patterns and Archetypes. In *19th Australian Conference on Software Engineering*, pages 338–346, 2008.

[15] Kirikova, M., Finke, A., and Grundspenkis, J. What is CIM: An Information System Perspective. In Grundspenkis, J., Kirikova, M., Manolopoulos, Y., and Novickis, L., editors, *Advances in Databases and Information Systems*, volume 5968 of *Lecture Notes in Computer Science*, pages 169–176. Springer Berlin / Heidelberg, 2010.

[16] Kleppe, A. G., Warmer, J., and Bast, W. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley, Boston, MA, USA, 2003.

[17] Meservy, T. and Fenstermacher, K. Transforming software development: an mda road map. *Computer*, 38(9):52 – 58, sept. 2005.

[18] Mili, H., Valtchev, P., Charif, Y., Szathmary, L., Daghrir, N., Béland, M., Boubaker, A., Martin, L., Bédard, F., Caid-Essebsi, S., and Leshob, A. E-Tourism Portal: A Case Study in Ontology-Driven Development. In *E-Technologies: Transformation in a Connected World*, volume 78 of *Lecture Notes in Business Information Processing*, pages 76–99. Springer Berlin Heidelberg, 2011.

[19] Molina, J. G., Ortín, M. J., na Moros, B., Nicolás, J., and Toval, A. Towards Use Case and Conceptual Models through Business Modeling. In *Conceptual Modeling - ER 2000*, volume 1920 of *Lecture Notes in Computer Science*, pages 655–663. Springer Berlin/Heidelberg, 2000.

[20] Nikiforova, O. and Pavlova, N. Open Work of Two-Hemisphere Model Transformation Definition into UML Class Diagram in the Context of MDA. In *Software Engineering Techniques*, volume 4980 of *Lecture Notes in Computer Science*, pages 118–130. Springer Berlin / Heidelberg, 2011.

[21] Object Management Group. *MDA Guide Version 1.0.1*, 2003.

[22] Object Management Group. *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*, 2009.

[23] Object Management Group. *Production Rule Representation (PRR)*, 2009.

[24] Object Management Group. *OMG Unified Modeling Language, Superstructure*, 2011.

[25] Prat, N., Akoka, J., and Comyn-Wattiau, I. An MDA approach to knowledge engineering. *Expert Systems with Applications*, 0(0):1–18, In press, 2012.

[26] Rodríguez, A., de Guzmán, I. G.-R., Fernández-Medina, E., and Piattini, M. "semi-formal transformation of secure business processes into analysis class and use case models: An mda approach". *Information and Software Technology*, 52(9):945 – 971, 2010.

[27] Rodríguez, A., Fernández-Medina, E., and Piattini, M. Towards Obtaining Analysis-Level Class and Use Case Diagrams from Business Process Models. In *Advances in Conceptual Modeling - Challenges and Opportunities*, volume 5232 of *Lecture Notes in Computer Science*, pages 103–112. Springer Berlin / Heidelberg, 2008.

[28] Said, I. B., Chaabane, M. A., and Andonoff, E. A Model Driven Engineering Approach for Modelling Versions of Business Processes using BPMN. In *Business Information Systems*, volume 47 of *Lecture Notes in Business Information Processing*, pages 254–267. Springer Berlin Heidelberg, 2010.

[29] Siegel, J. and OMG Staff Strategy Group. Developing In OMG's Model-Driven Architecture, November 2001. Available at: `http://www.omg.org/mda/mda_files/developing_in_omg.htm`.

[30] Siqueira, F. L. and Muniz Silva, P. S. Mapping the WRSPM Model to Model-Driven Architecture Models. In *Proceedings of the Eighth International Conference on Information Technology: New Generations*, pages 750–753. IEEE Computer Society, 2011.

[31] Touzi, J., Bénaben, F., and Pingaud, H. Prototype to Support Morphism between BPMN Collaborative Process Model and Collaborative SOA Architecture Model. In *Enterprise Interoperability III*, pages 145–157. Springer London, 2008.

[32] Valderas, P., Fons, J., and Pelechano, V. From Web Requirements to Navigational Design - A Transformational Approach. In *Web Engineering*, volume 3579 of *Lecture Notes in Computer Science*, pages 557–575. Springer Berlin / Heidelberg, 2005.

[33] Van Lamsweerde, A. Goal-Oriented Requirements Engineering: A Guided Tour. In *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering*, RE '01, pages 249–262. IEEE Computer Society, 2001.

[34] Weber, G. Technology-independent modeling of service interaction. In *Proceedings of the 2008 12th Enterprise Distributed Object Computing Conference Workshops*, EDOCW '08, pages 35–42, Washington, DC, USA, 2008. IEEE Computer Society.

[35] Zave, P. and Jackson, M. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6:1–30, January 1997.

[36] Zdravkovic, J. and Ilayperuma, T. A Model-Driven Approach for Designing E-Services Using Business Ontological Frameworks. In *14th IEEE International Enterprise Distributed Object Computing Conference (EDOC)*, pages 121–130, 2010.

[37] Zhang, L. and Jiang, W. Transforming Business Requirements into BPEL: A MDA-Based Approach to Web Application Development. In *IEEE International Workshop on Semantic Computing and Systems*, pages 61–66, 2008.

[38] Zhang, W., Mei, H., Zhao, H., and Yang, J. Transformation from CIM to PIM: A Feature-Oriented Component-Based Approach. In *Model Driven Engineering Languages and Systems*, volume 3713 of *Lecture Notes in Computer Science*, pages 248–263. Springer Berlin / Heidelberg, 2005.