International Journal of Engineering Inventions
e-ISSN: 2278-7461, **p**-ISBN: 2319-6491
Volume 2, Issue 2 (January 2013) PP: 83-88

Different Proposed Models to Mapping MDA to RUP

Mehdi Yaghoubi[1], Mahmood Yaghoubi[2], Manoochehr Babanezhad[3]
[1] Department of Computer Sciences, Faculty of Sciences, Golestan University, Gorgan, Golestan, Iran
[2]Department of Computer Sciences, Mirdamad High Education Institute, Gorgan, Golestan, Iran
[3] Department of Statistics, Faculty of Sciences, Golestan University, Gorgan, Golestan, Iran

Abstract:- Different methods such as object-oriented, component-oriented, aspect-oriented, and service-oriented are proposed to establishing Software engineering over the past decades. Model driven development is one of the methodologies that had considered recently and is growing rapidly. Reducing of the production expenses and preventing software intricacies is one of the software engineering aims modeling the program logic and then changing model to a practicable code by some tool automatically will be the aim of the model driven development. In this paper we investigate some model driven methodology and also we'll preferred MDA mapping to RUP.

Keywords:- Model Driven Development, Model Driven Methodology, Model Driven Architecture, RUP.

I. INTRODUCTION

MDA response to the challenge of production software application companies were faced it during recent years; one of the questions that might be likely for a software engineer application after several years of activity in this field is formed is that you can't use design and source code for previous projects in the new project, How to previous projects is useful for future work as capital and reservoir? The other question, in the software based organizational and they are used by users how can changes platform by least inconvenience and cost?

MDA says: The ideal case for a software factory could happen when are designed the program logic model and then some tools convert them to executable code models are not changed and the details of implementation are making separate. MDA is an approach that was introduced by the object management group and is standard. A style of MDA Software Development based on the use of automated tools for building independent models of the system and converts them to effective implementation [1]. The rest of the paper is organized as follows: section 2 we will overview of MDA, section 3 MDA effects on the RUP methodology are investigated and investigate MDA methodology in chapter 4 and conclude in chapter 5. MDA Models are as follows: Calculation independent model (CIM), Platform-specific model(PSM) and Platform-specific implementation(PSI) [10].

II. MDA EFFECTS ON THE RUP

Using MDA in RUP there changes in [11] to express these changes is discussed (Figure 1). At the Inception phase, requirements for the system are being elicited, resulting in a CIM. A CIM model covers the Business Modeling and Requirements disciplines. At the same time, during the Inception phase, metamodel planning for the PIM development and initial implementation are carried out. Elaboration is the main phase impacted by an MDA project. It is important to look at the Elaboration activities and briefly describe the MDA modifications. The Architect role is the main role in the Elaboration phase that requires additional consideration. As a specialisation of "Architect", the MDA Architect role is appropriate for many MDA projects. Essentially, this role defines specific MDA activities and artifacts, creates the transformations, and so on. The primary MDA artifacts of this role are mapping documents, transformations, and UML profiles. PIM also covers some parts of the Construction – the models should be enriched with mapping functions and appropriately marked. Models are iteratively tested for conformance. At the Construction phase, model transformation into different PSM or codes takes place. PSM covers the Transition phase and Deployment discipline as well – transition to some production environment can be performed with the help of a separate PSM as well. The Configuration and Change Management discipline is also affected by the MDA – these disciplines cover metamodel and metadata repository maintenance. Typically, the MDA automates activities within the RUP. Rather than changing RUP activities, the MDA enhances them with additional tasks aimed at supporting automation with a number of the primary RUP activities. The primary changes to the RUP involve a more subtle change of perspective in the development process. MDA encourages architects and developers to work at higher levels of abstraction than typically expected in non-MDA projects. This is most apparent during construction, where the code automation

aspects of MDA significantly change the emphasis of the implementation tasks. They work less with actual implementation models and source codes themselves, and more with designs for the appropriate business-focused workflow of the solution. A smaller subset of developers will implement the model-to-code transformations themselves.

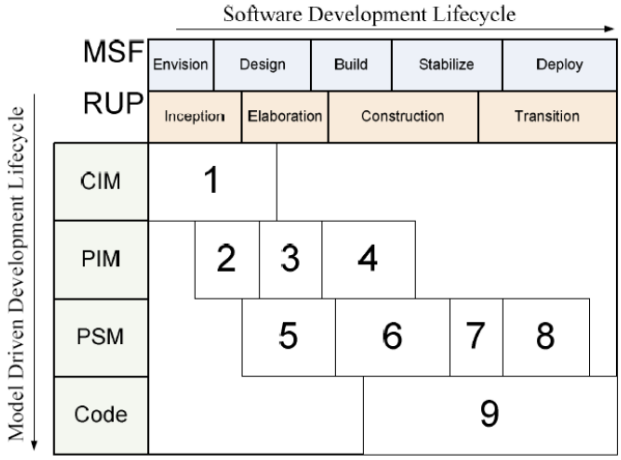

Fig 1. Software development lifecycle and MDA mappings

More detailed descriptions of Figure 4 in Table 1

Table 1. MDA specific activities in the phase crossing

1	Define the possible domain specific metamodel, Traceability convention from requirements to code, Choose the modelling tool, Technology selection, Requirements specification development (ontology), Information structuring, Initial PIM development
2	Transformations strategy selection, Business process modelling role evaluation, Risk assessment
3	Refine the PIM context, Define verification strategies, PIM design
4	Finalise metamodels, Create model-capturing support, Annotate the PIM with platform specific information, Verify the models
5	PSM identification, Define the verification strategies, Define the user-interface options, PSM metamodel design
6	Verify the models, Mark the PIM and PSM, Create additional transformations and code generation scripts for target platforms, Create test scripts, PIM to PSM automatic or semi-automatic transformation
7	Verify the established meta-models, Code files generation from PSM
8	Automatic deployment
9	Fine-tuning and repeated use of the automated testing process, Executable code verification to models

III. SOFTWARE DEVELOPMENT METHODOLOGY BASED ON MDA

3.1. MODA-TEL

MODA-TEL is proposed as a software development process based on MDA principles and concepts [2, 3, 9]. It is specialized for distributed applications, but is general enough to be applicable to other domains and situations as well. The MODA-TEL process is defined in accordance with OMG's Software Process Engineering Meta-model (SPEM). MODA-TEL separates preparation activities from execution activities in distinct phases. The following phases are identified by this methodology (figure 2).

1. Project Management: During this phase the software development process is selected and described in terms of its activities; identified activities are then allocated to roles, and procedures for quality assurance are put into place.

2. Preliminary Preparation: The objective of this phase is to identify modeling and transformation needs. The final execution platform of the system is also identified and expressed as an *abstract platform*. The appropriate

modeling language for the project and its specific needs are also identified. Required transformations between models are specified based on the selected modeling languages.
3. Detailed Preparation: Models and model transformations are specified in this phase. The specifications of modeling languages and model transformations are prepared according to the needs identified in the previous phase.
4. Infrastructure Setup: In this phase, tools are selected for supporting MDA- based development activities. For example, tools for automatic/semi-automatic code generation may be selected. The metadata management facility is also defined in this phase.
5. Project Execution: This phase spans the main software development activities. Activities of this phase depend on the software development process selected in the project management phase. Seven activities in this phase: *requirements analysis*, *modeling*, *verification and validation*, *transformation*, *coding and testing*, *integration and deployment*, and *operation and maintenance*.

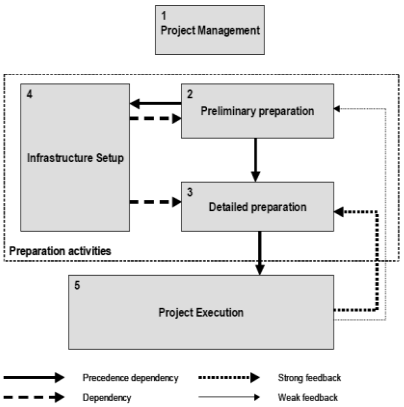

Fig 2. Phases of the MODA-TEL methodogy

3.2. MASTER

MASTER is a European IST project during which a MDA-based methodology [8]. The process consists of eight main phases, each of which includes various activities. The phases are:
Capture User Requirements: The objective of this phase is to elicit and document customer requirements. Products which are produced during this phase are: an Application Model in which customer requirements are formalized, an initial Application PIM, and an initial functional requirements specification.
PIM Context Definition: The system goals as well as the scope of the system are defined in this phase. Other activities of this phase are: identifying the external actors of the system, specifying the main services offered by the system, and identifying the business objects exchanged between actors and the system.
PIM Requirements Specification: The main activity in this phase is refining the PIM Context produced in previous phase, as well as specifying use cases, and identifying non-functional requirements as well as modeling their relationship with functional requirements.
PIM Analysis: In this phase, system functionalities and QoS aspects are described with a view to the interior of the system.
Design: In this phase, a platform-independent design is first performed for all the requirements. The design is then refined in order to denote the platform-specific solution.
Coding and Integration: According to the ideal MDA approach, the code is to be produced automatically from the PSM through transformation engines.
Testing: Test cases are to be generated automatically from the test model.
Deployment: In this phase, the developed system is delivered to the customer.

3.3. MIDAS

MIDAS is a model driven methodology framework of Web Information Systems (WIS) [5, 12]. UML is used for representing the different PIMs and PSMs which are proposed in this framework. MIDAS focuses on three dimensions of modeling a WIS.
1) *Levels*, which refer to the hypertext content and the resentation levels.
2) *Phases*, which refer to the phases of the software lifecycle.
3) *Aspects*, which refer to the structural and behavioral modeling viewpoints. Platform Independent Models, Platform Specific Models, and Computation Independent Models as well as the relevant mapping rules are defined according to these dimensions.

3.4. C^3

C^3 is a software development process which is embedded into a concurrent, collaborative and component-based methodology enriched with MDD techniques [6]. The main feature of the C^3 architecture is its domain repository which contains domain-specific metadata and components. Its process consists of two Main phases (figure 3):

Standardization Phase: In this phase, domain software assets which are archived in a repository are accessed and downloaded onto the *project repository.*

Software Development Phase: This phase includes three main steps which are: 1) *Model Design*, in which component developers select a business application architecture from the project repository to work on. 2) *Code Generation***,** which is performed after modeling the business application with UML or XMI on a platform independent level. Code generation tools transform the models into platform-specific software components. 3) *Application Deployment*, in which components are deployed into the user environment based on the architectural framework designed.

3.5. ODAC

ODAC is a methodology based on the Reference Model of Open Distributed Processing (RM-ODP) [4] with the potential to be a MDA-oriented methodology The ODAC process consists of three main phases (figure 4).

1) *Analysis*, in which the *Behavioral Specification* – a PIM describing the system according to its objective and its role in the business – is produced; 2) *Design*, in which the *Engineering Specification* – a PDM that is the description of the execution environment – is produced; 3) *Implementation*, in which the *Operational Specification* – a PSM that is the result of the transformation of the PIM as configured according to the PDM – is produced.

ODAC prescribes steps for each phase as well as a number of guidelines for producing the relevant artifacts.

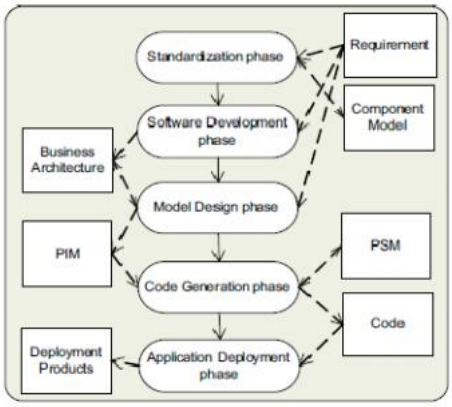

Fig 3. The C3 Process

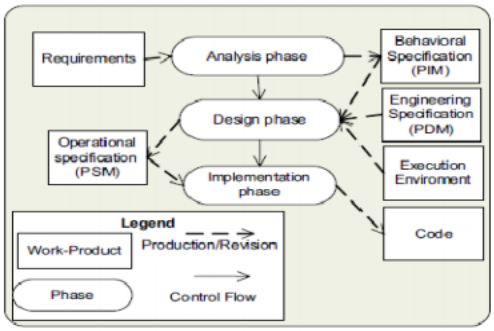

Fig 4. The ODAC Process

3.6. DREAM Methodology

The Dramatically Effective Application development Methodology (DREAM) combines the key activities of Product Line Engineering (PLE) with the model transformation features of MDA [7]. DREAM phases are as follows (figure 5)

Domain Analysis: captures the features of several organizations in the same domain, and analyzes the Commonality and Variability (C&V). The output is the specification of common features and differences between organizations.

Product Line Scoping: determines the scope of the target product line.

Framework Modeling: realizes the C&V in a framework, presented as a PIM. The framework defines the general architecture for the desired members of the product line, together with the relationships and constraints.
Application Requirements Analysis: analyzes the application requirements and identifies the features related to the application at hand. The output of this phase is the application analysis model.
Application-Specific Design: realizes the application analysis model as a platform-independent design model. The output is the application-specific PIM.
Framework Instantiation: instantiates the framework for the specific application by setting the variants accordingly. The output of this phase is the instantiated framework PIM.
Model Integration: integrates the specific application PIM and the instantiated Framework PIM into one model.
Application Detailed Design: refines the integrated model by considering platform-specific issues, thereby producing the PSM.
Application Implementation: generates the execution code and its related implementation complements – such as the database – from the PSM.

IV. CONCLUSIONS

Although many have invested on the MDA, backup methodology to develop the model-based system is under fading. We consider a few methodologies of MDA specially. In this study, we found MDA support all of model-based methodologies also in considered methodology, MASTER covered general life cycle of software development completely but MODA-TEL, ODAC, C3 and DREAM partially covered them and MIDAS do not covered MDA life cycle, details of this comparison are showed by table 2, also in this article we analyzed mapping of MDA life cycle on RUP life cycle.

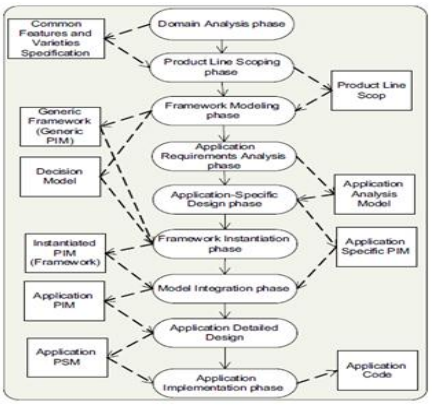

Fig 5. The DREAM Process

Table 2: comparison life cycle phases of MDA methodology

	MODA-TEL	MASTER	MIDAS	ODAC	C³	DREAM
requirements	B	C	A	A	B	B
Analysis	B	C	A	C	A	B
Design	B	C	A	C	B	B
Implementation	B	B	A	B	B	B
Test	B	C	A	A	A	A
Deployment	B	B	A	A	B	A
Maintenance	B	C	A	A	A	A
A: The methodology does not provide coverage for the phase. B: The methodology provides general guidelines for the phase. C: The methodology provides detailed directives for the phase.						

REFERENCES

1) Booch G. , Brown B. , Iyengar S. , Rumbaugh J. , Selic B. (2004).*An MDA Manifesto*. MDA Journal.
2) Gavras .A., Belaunde .M., Ferreira Pires .L., and Almeida.J.P.A. (2004). *Towards an MDA-based development methodology for distributed applications.* In Proceedings of the 1st European Workshop

on Model-Driven Architecture with Emphasis on Industrial Application (MDAIA). University of Twente, Enschede, The Netherlands, pp. 43-51.

3) Gavras .A., Belaunde .M., Ferreira Pires .L., and Almeida.J.P.A.(2004).*Towards an MDA-Based Development Methodology*. *EWSA*.
4) Gervais, M. ODAC.(2003). *An Agent-Oriented Methodology Based on ODP*. Journal of Autonomous Agents and Multi-Agent Systems, 199- 288.
5) Hildenbrand. T, and Korthaus.A. (2004). *A Model-Driven Approach to Business Software Engineering*. Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI), Volume IV Information Systems. Technologies and Applications: I, IIIS. Orlando. Florida. USA. July 18-21.
6) Hildenbrand, T., Korthaus, A.(2004).*A Model-Driven Approach to Business Software Engineering*. 8th World Multi-Conference on Systemics, Cybernetics and Informatics, Florida. Information Systems, Technologies and Applications, vol.~IV, pp. 74—79.
7) Kim, S., Min, H.G., Her, J.S., Chang, S.H. "DREAM: A practical product line engineering using model driven architecture" ICITA 2005, Australia, pp.70-75(2005).
8) Larrucea .X, Diez .A.B.G, and Mansell.J.X. (2004).*Practical Model Driven Development Process*. Technical Report, NUMB 17,University of Kent, , pp. 99-108.
9) MODA-TEL project. (2003). Deliverable D3.2, Guidelines for the application of MDA and the technologies covered by it. http://www.modatel.org/public/deliverables/D3.2.htm.
10) Ostadzadeh, S.S., Shams, F., Ostadzadeh, S.A. (2008). *An MDA-Based Generic Framework to Address Various Aspects of Enterprise Architecture*. Advances in Computer and Information Sciences and Engineering. Springer. pp. 455–460.
11) Parviainen. P., Takalo.J, Teppola .S. & Tihinen.M. (2009). *Model-Driven Development Processes and practices*. VTT WORKING PAPERS.
12) 12-Web Information Systems Development.(2004). International Conference on Enterprise Information Systems (ICEIS).523-526.