

Enhanced Model Driven Architecture Software Development Life Cycle with Synchronized and Consistent Mapping

Atika Qazi¹, Saif Ur Rehman Malik¹, Madiha Kazmi¹, Shams-Ur-Rehman², Nouman Maqbool Rao³

¹Department of Computer Science
COMSATS Institute of Information
Technology, Islamabad, Pakistan
{atika_qazi, madiha_kazmi@comsats.edu.pk }, meet_saif@hotmail.com

²Department Computer Science
International Islamic University
Islamabad, Pakistan
shams_ur_rehman@hotmail.com

³Department of Monitoring and Evaluation
Higher Education Commission Pakistan
nmaqbool@hec.gov.pk

Abstract - Model Driven Architecture (MDA) is an approach for specifying systems in terms of models, and then enacting the development process through performing model transformations. The System Development Life Cycle (SDLC) is a process for ensuring that all functional and user requirements and agency strategic goals and objectives are met. The SDLC provides a structured and standardized process for all phases of any system development effort. A generic lifecycle for MDA-based software development that can be used as a basis for constructing MDA-based methodologies through a Method Engineering (ME) process has been proposed. This generic life cycle is studied and analyzed in this paper. The MDA-SDLC has mapped with traditional SDLC using it as benchmark. Shortcomings in MDA-based SDLC are identified. This study will lead to a more generic and standardize MDA-based SDLC.

Keyword: Model Driven Architecture (MDA), System Development Life Cycle (SDLC), Model Engineering (ME), Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM)

1. I. INTRODUCTION

The MDA [1] is a new way of developing applications and writing specifications, based on a platform-independent model (PIM) of the application or specification's business functionality and behavior. A complete MDA specification consists of a definitive platform-independent base model, plus one or more platform-specific models (PSM) and sets of interface definitions, each describing how the base model is implemented on a different middleware platform. A complete MDA application consists of a definitive PIM, plus one or more PSMs and complete implementations, one on each platform that the application developer decides to support. MDA development focuses first on the functionality and behavior of a distributed application or system, undistorted by idiosyncrasies of the technology platform or platforms on which it will be implemented. In this way, MDA divorces implementation details from business functions. The SDLC is System Development Life Cycle [2], defined by many other ways by different authors. The SDLC suggests the use of a step by step approach to develop a system. The SDLC is having six to eight stages but it vary between them by different proposals, Here SDLC is with six stages the stages defined to progress step by step. In order to improve the development process and produce quality product SDLC allows the analyst to

proceed step by step for the analysis and design of the information system. The analyst must start from stage first and proceed next simultaneously. Each stage having questions, analyst has to answer the questions to achieve the result in successful way. Each step performs useful activities, although the activities are separate from each other in a sequence but in real sense they are highly interrelated. Activities in each stage are isolated but if in a stage problem occur then it might require change by the analyst in previous stage to solve it. Analyst is person that must be able to tackle the problem in systematic way by using his/her skills under certain set of application tools and techniques. Therefore each stage is defined separately to perform specific activity but interrelated as the quality results cannot be generated without the interaction among them. It is more appropriate to say that activities in SDLC are performed in full swing overlap each other to make a useful quality end product. Prominent MDA-based methodologies are MODA-TEL [3], MASTER [4], C3 [5], ODAC [6, 7], DREAM [8], and DRIP-Catalyst [9]. In [10], a generic lifecycle for MDA-based

Software development has been proposed that can be used as a basis for constructing MDA-based methodologies through a Method Engineering (ME) process. However, this generic life cycle is silent about some important activities. This work identifies the shortcomings in MDA-SDLC by comparing it with traditional SDLC. This work will lead to a new more generic MDA SDLC.

The organization of this paper is as follows: section 2 describes MDA; section 3 maps MDA-SDLC with traditional SDLC i.e. with general system development lifecycle. In section 4, short comings of MDA-SDLC are described. Section 5 concludes the work and contains future work as well.

2. II. MODEL DRIVEN ARCHITECTURE (MDA)

The MDA is a model driven architecture. The MDA is a software development framework defined by OMG based on automatic transformations of models. By using modelling languages as declarative programming languages the result produced are raising of abstraction level for programmers and improved productivity, quality, longevity outlooks. The MDA is brought up to preserve investment in case if platform changes it should not effect technology. MDA provide the solution by isolate information and processing logic from technology specifics. MDA builds platform independent models like UML is independent of COBRA, COM, EJB, XML etc. The MDA maintain the separation at the implementation level.

The MDA application involves Platform Independent Model (PIM), Platform Specific Model (PSM) and code. To represent different model multiple tools are used, like Unified Modeling Language (UML) Meta-Object Facility (MOF) and Common Warehouse Meta-model (CWM) etc. MDA is model-driven because it provides a means for using models to direct the course of understanding, design, construction, deployment, operation, maintenance and modification. MDA Based methodologies also used Situational Method Engineering (SME). The use of SME is to fit the project according to certain situation. There are different types like Paradigm based SME, Generic instantiable process lifecycle and the Extension based approach. The detail of CIM, PIM, PSM and code generation is given below:

2.1. A. 2.1 CIM

The CIM captures the business knowledge without reference to a particular system implementation or technology, and it would remain the same even if the system were implemented mechanically, or in computer software. The CIM is described with a vocabulary that is familiar to business people. Therefore, it plays an important role in bridging the gap between experts about the business, and software engineers.

2.2. B. PIM

The platform independent model is responsible for formal specification of the structure and function of a system excluding the platform detail. Usually UML is used to represent it sometime any other tool can be used instead of UML.

2.3. C. 2.3 PSM

The platform specific model is responsible for the functionality in a particular platform that is specified in PIM in general. It takes PIM as its input and transforms PIM into PSM. It uses UML to make Platform specific model. Mapping is carried out at this stage for transformation of PIM into PSM according to desire platform. The PIM provide the benefit of verification according to requirement before transforming in into PSM.

2.4. D. 2.4 Coding

At this stage coding is finalized. An MDA tool generates most of the implementation code selected by the developer and some part is done manually by the developer. Finally the end product is produced.

3. III. MAPPING OF SDLC WITH MDA-SDLC

The generic life cycle focused on the process part of methodology. The generic SDLC consist of five phases and activities in them. Each phase in a sequence have its own activity to perform. Project Initiation Phase, PIM development Phase, PSM and Code development phase, Deployment Phase, Maintenance Phase are the phases of having phases and activities in them. The mapping of MDA-SDLC with SDLC is given below. MDA-SDLC is not a solid methodology but it's a general process

Table1: Mapping of SDLC with MDA-SDLC

| | |
|--|--|
| 1) Recognition of need preliminary survey/initial investigation | Phase 1) "Project Initiation" 1) CIM definition |
| 2) Feasibility Study | 2) Requirements Specification 3) Obtaining Funding & Support (Resources) 5) Define General Plan |
| 3) Analyzing | Phase 2) "PIM Development" 1) Produce Analysis PIM (System Functionalities) 2) Architectural Design |
| 4) Design | Phase 3) "PSM & Code development Phase" 2) Generate Code 3) *Testing |
| 5) Implementation | Phase 4) "Deployment" 1) Transition of the system to the user environment. 2) *Finalizing of the system and user documentation. |
| 6) Post-Implementation | Phase 5) Maintenance 1) *Maintenance |

4. IV.SHORTCOMINGS IN MDA-SDLC

Mapping of MDA-SDLC with traditional SDLC shows that following activities/phases are not included in the MDA-SDLC i.e. these are the major shortcomings of MDA-SDLC.

4.1. A. Modification and Enhancement Phase

System should have the capacity for modifications if required not only the maintenance feature is enough to make system efficient and reliable. Such as if future enhancement is required or there is a need for up gradation, the system should be flexible enough to absorb the changes. Enhancement must take place in such a way that by adding extra features with in the body of current system sequential consistency should not be disturbed. Organizational requirements can change with times which demand for system enhancement.

4.2. B. Training to User

Developers adopt various approaches to make the system easy and understandable for the common user. Documentation is almost always provided to the user along with the system there are instructions to use the system and for troubleshooting, but the much better approach is the practical demonstration of the system by the developer. Such presentations and demos delivered by the developer make system very easy to use and manage troubles in more handsome way.

4.3. C. Quality Check and Evaluation after each Phase

There must be evaluation after regular intervals. Overall evaluation after each phase must be carried out by feed back from user to confirm that development process is going according to his requirements and to make sure that development team is on right track and working according to defined standards.

In order to make quality product senior supervisor must be responsible for conducting demo to understand the current project for new team member .Quality of software shout not be compromised by any disorder during development. Quality check activity after the first phase of MDA-SDLC to make sure that what the user required developer understand its well, and what developer team is going to propose is that which user demands for. The quality check activity will be useful to confirm that the requirement model is meeting the standard; project infrastructure is also up to the mark, the sources for the project funding will also be available in near future and over all base information for PIM is from valid sources and is that which is required.

5. V. CONCLUSION

In this paper traditional SDLC is taken as benchmark. MDA-SDLC is mapped with traditional SDLC. It is found that the MDA-SDLC lack some very important phases or activities such as modification and enhancement phase, user training phase and Quality check and evaluation phase .In other words, MDA-SDLC requires to be modified and enhanced. As future work, a new, more generic MDA-SDLC will be proposed and evaluated.

6. REFERENCES

- [1] I. Mukerji, J. Miller, MDA Guide Version 1.0.1, OMG, 2003. [2]<http://www.oft.state.ny.us/pmmp/guidebook2/phase.pdf>
- [2] A. Gavras, M. Belaunde, L. Ferreira Pires, J. P. Andrade Almeida, "Towards an MDA-based development methodology", Proc. First European Workshop on Software Architecture (EWSA2004), Enschede, Netherlands, 2004, pp. 71-81.
- [3] X. Larrucea, A. B. G. Diez, J. X. Mansell, "Practical Model Driven Development process", Proc. Second European Workshop on Model Driven Architecture (MDA), Canterbury, UK, 2004.
- [4] T. Hildenbrand, and A. Korthaus, "A Model-Driven Approach to Business Software Engineering", Proc. 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004), Volume IV Information Systems, Technologies and Applications, Orlando, Florida, USA, 2004, pp. 74-79.
- [5] M. Gervais, "ODAC: An Agent-Oriented Methodology Based on ODP", Journal of Autonomous Agents and Multi-Agent Systems, 7(3), 2003, pp. 199-228.

- [6] M. Gervais, "Towards an MDA-Oriented Methodology", Proc. 26th Annual International Computer Software and applications Conference (COMPSAC'02), Oxford, England, 2002.
- [7] S. Kim, H. G. Min, J. S. Her, S. H. Chang, "DREAM: A practical product line engineering using model driven architecture", Proc. ICITA '05, Australia, 2005, pp. 70–75.
- [8] N. Guelfi, R. Razavi, A. Romanovsky, S. Vandenberg, "DRIP Catalyst: an MDE/MDA Method for Fault-tolerant Distributed Software Families Development", Proc. OOPSLA & GPCE 2004 workshop on best practices for Model Driven Development, 2004.
- [9] Mohsen Asadi, Mahdy Ravakhah, Raman Ramsin, An MDA-based System Development Lifecycle, Second Asia International Conference on Modelling & Simulation, Kuala Lumpur, Malaysia 13 – 15 May 2008