Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 8, July 2021: 585-592

Understanding Model Driven Architecture: Framework and Tools

Simmi Bagga[1]and Dr. Anil Sharma[2]

[1, 2] School of Computer Applications, Lovely Professional University, Punjab, India

simmibagga12@gmail.com
anil.19656@lpu.co.in

Abstract

Model Driven Architecture (MDA) is a technique that permits the developer to build application models that are platform independent. It allows the developers to build models of a project without the specific in-detail knowledge of applications to be involved and then combining those models to create the application. The main idea of MDA is to represent the business logic in the form of abstract models. These abstract models are mapped into different platforms by applying a set of transformation rules. The models are usually described by UML diagrams in a formalized manner, which can be used as input for tools that perform the transformation process.It defines a domain-specific language (DSL) to be used along with a platform-independent model (PIM). In this paper, the authors will discuss what is MDA, what models are used in it, what these models mean in terms of practical applications and also the common tools available in the market to implement MDA.

1.0 Introduction

The process of software development is a topic that has been put into the limelight by many researchers since the start of the 21st century. Object Management Group (OMG) leads these researchers in developing a framework model that is most stable, easy to use and has a wide range of applications. In 1996, OMG planned to include modeling in the scope of Software Development and thus adopted Unified Modeling Language (UML) and MetaObject Facility (MOA) [1]. Still, these models were not directly relatable to the software product. Model Driven Architecture was thus introduced to act as a middleware between these standardized developmental models and the software product. MDA approach was designed with the intent of fully incorporating the benefits of standard models such as interoperability, portability, and reusability in the software development lifecycle.The use of MDA technique is to increase the interoperability and maintainability [1][4].In MDA, codes are produced from user given models or logical diagrams, hence a forward engineering approach is followed. A model that fits the specific business logic is either prepared or derived from an existing model. MDA provides an architecture that supportsplatform independence, interoperability across different platforms, productivity, domain specificity, andportability. In tnis approach, system functionality's specifications (done in PIM) and implementation (done in PSM) are separated from each other as they are done in two different phases. The system requirements are specified in the Computational Independent Model (CIM). Another benefit of MDA is that fewer efforts are needed to develop the whole system thus improving productivity [2][3].

MDA approach focuses on developing the highest level of abstraction models of the business logic and further promoting to other levels by transformation of one model to another. MDA approach classifies four types of models that are used while the development of software. These are CIM (Computational Independent Model), PIM (Platform Independent Model), PSM (Platform Specific Model) and Code.

CIM (Computational Independent Model)

CIM is a simple representation of a system that is understandable by a layman without specific information about how it is to be implemented. It only shows the business logic of the system represented in the form of models. According to MDA guidelines, CIM has to be developed such that it can further contribute to the development of PIM and PSM[2][4].

PIM (Platform Independent Model)

PIM is the view of a system in a greater elaboration. More details are included but the system is still kept, platform independent.It contains information about business functionalities and procedures' algorithms but nothing about the technical stack or the particular platforms on which it has to be implemented. Virtual machines that are technology independent are often used to implement PIM because they are easily extensible to PSM [1, 5].

PSM (Platform Specific Model)

PSM is a view of a system that focuses its implementation on a particular platform. A PSM is a result of the translation of PIM by using guidelines and working procedures of the used platforms. It gives information that all functions are to be done in each platform used, how to provide connectivity between the used platforms, how data travels throughout one platform and further to other platforms used etc. Thus it becomes very easy to generate high-level program codes from PSM. It must be noted that multiple PSMs can be created from a given PIM by changing the set of technical stack[2][4].

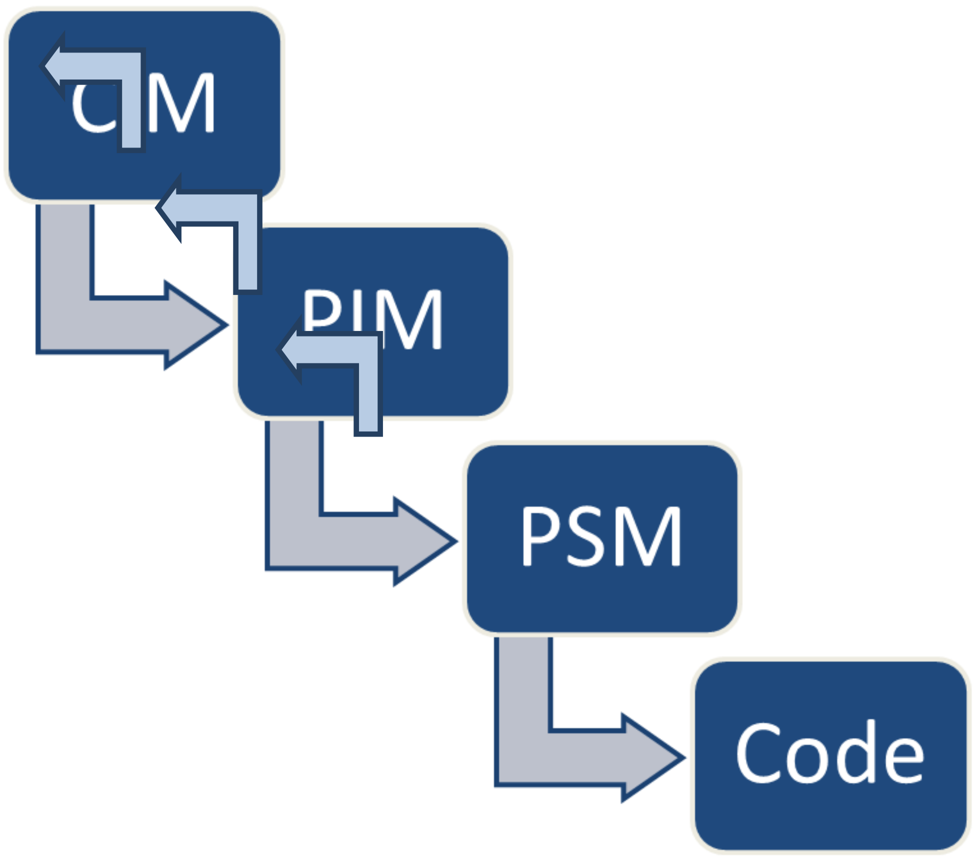

Fig1: Different levels of MDA

Model transformation,as the name suggests, is the process of mapping one model to another model. A model transformation system takes a system model as an input along with some other information and produces another model as output.This can be from platform independent to platform specific models or vice versa. This transformation can be done in many ways. Whichever way it is done, it produces, from a PIM, a model-specific to a particular platform.

The transformation is completed by the mapping process. Mapping provides guiding principles for the transformation of PIM to PSM. Various approaches are using for mapping.One of them is Model type mapping in which it specifies model built using types defined in the PIM language are transformed into types defined in the PSM language. In this type, the mapping gives rules for the transformation of all instances of types in the metamodel of the PIM language into instances of types in the metamodel of the PSM languages[1,2,4,5].

Example Applications of MDA

1.) Consider an application development project in which there is a need to develop a library management system. This may have features common to any library such as functionalities for searching of books, borrowing and returning of books, fine calculation for late return with payment gateway, book bank facility with records stored in database and records of users.

For simplicity, it would be better to look only at one part of this application, say fine collection. Based on the MDA approach, the first step will be, to begin with, CIM. CIM for this functionality will consist of UML diagrams depicting the features of these functionalities such as increasing the fine by a certain amount after the due date, which user's fine is to be increased, set the amount payable to zero after fine is paid, generate notifications to users and other such features as per the requirements. PIM will show how to retrieve data from the database and display to users about their fine; the steps involved in completing the transaction will be mentioned but only in general. However, in PSM, all these steps will be detailed. A base for high-level programming will be developed keeping in mind that what all technical stack is to be used in the project. Thus MDA representation of the project will be completed [1, 2].

2.) As it is seen, from the last few years, the sources of data have become heterogeneous and massive. It is very difficult to manage such data and use it for decisional system. Several recent research works have focused on the architectures of data warehouses for this kind of datastores i.e. relational and non-relational (NoSQL). Major research in this domain is intended on the implementation of new algorithms for interrogation of these warehouses while improving their response time. If these data warehouses are realized using traditional models, then migration of information from one kind of datastore to another is very difficult. On the other hand, MDA provides an architecture that simplifies the interoperability between different datastores resulting in the reduction of efforts [4][6].

2. Literature Review

Yashwant Singh, Manu Sood discussed various structural models likes CIM, PIM & PSM and also explained rules of transformation from different levels i.ePIM to PSM. For the software development process of MDA, it is necessary to have clearly defined techniques for the definition and execution of automated model transformation. Authors also discussed transformation through real-life example. The authors concluded that by using the MDA technique Platform dependent code can be automatically generated from the Platform Independent Model. MDA can easily handle the challenges of changing technologies, multiple platform environments, interoperability, etc. [1].

AtifAftab et.al explained different approaches of Model Transformation. The heart of model driven architecture is its transformation. These are mainly two types of transformation of MDA i.e. Model to Model and Model to Code Transformation. Different parameters have been identified and applied to the existing techniques and found various transformation approaches lacks in terms of tool support only very few models to model transformation approaches have tool support. Most of the transformation approaches are stateless and unidirectional and most of the approaches use only the UML class diagram for specifying transformation [2].

Yashwant Singh, Manu Soodexplained the MDA software development approach can improve various factors of information systems like improve quality, longevity, cost of production &

success rate of an information system, etc. by separating the concern through abstraction at various levels. The authors explained the notion of various models, transformation among models, and the software development life cycle process of MDA. The authors also explained Computational Independent Model (CIM) informs software developers about structured& dynamic aspects of a system [4].

Abdelhediet. al. proposed an automatic approach that guides and facilitates the Big Data implementation task with NoSQL system based on MDA. The proposed approach provides a set of transformations that generate a NoSQL physical model from a conceptual model. The authors also proposed different solutions to transform the binary relationships of the logical model under Cassandra and MongoDB [5].

H. Dev, A. Seth explained a huge amount of data is collected in banks that are unable to maintain and utilize it in a proper manner and also faces the problem of reusability. Authors proposed that using MDA the design of the banking database helps to improve various features like maintainability, portability, and flexibility, interoperability, etc. Authors also proposed that MDA can be used to represent system specifications, which separates the specification related to implementation that helps to react quickly by changing functional and technological requirements [6].

Fabio Perez Marzullo et.al. measured the performance of a database. Due to component based development, performance issues need to be checked thoroughly. As most IT projects are based on model approach, we need to check performance and identify faults while construction of those projects using Model based Approach. The authors proposed an MDA extension that enables code generation to conduct a performance analysis. They used the AndroMDA tool, profiling libraries JAMon and InfraRED, and creates profiling cartridge to generate assessment code [7].

A. Rosa et.al.proposed an approach for database modeling using MDA. This approach facilitates the designer to choose between different conceptual modeling notations to automate the entire process. Authors also discussed a tool that is more integrated with different Database Management Systems, contributing to the versatility over the development and increase the reusability of the models [9].

S. Kumar Mishra et.al.explained the design of the HealthCare System using the MDA approach. The system designed using the MDA approach provides benefits like interoperability, portability, reusability, and maintainability. Different levels of MDA modeling like CIM, PIM, and PSM are discussed concerning the Healthcare System [10].

3. Tools Of MDA

There are a lot of MDA tools today available in the market. Some of these are given below:

AndroMDA

AndroMDA is an open source framework that follows the MDA paradigm. It’s a plug-in architecture in which it combines several plugins and produces several components that may be swapped in and out at any time. While developing any application only need to develop plugins related to that and then plugins helps to generate components for any language. The existing projects can be used for both platform-specific purposes and general services [12].

The developer of AndroMDA can easily extend the existing modeling language by using the “metafacades”. The extension in existing is done by using the modeling libraries and templates available in transformation tools. The main focus of AndroMDA is to generate abundant code from PIM by using UML. It does not provide the facility of PSM inspection and manipulation between PSM and PIM. The main reason behind this is the trade-off between the complexity of PIM and PSM traceability and the benefits of maintaining PSMs for different platforms. At the level of UML, this technique works well as it only involved general platform independent semantics. The main benefit of using the AndroMDA is it eliminates the need to write code again and again and the project model reproduces the code. Projects are documents in PIM and can convert easily to changing technology i.e. it adapts very fast according to the changing technology [12][14].

ArcStyler

Arcstyler is one of the prominent software development tools for MDA that supports the J2EE and .Net platform and has the feature of cross platform and providesa standard environment. It fully supports java for designing highquality application of any size. It enables the developer to generate applications in an automated manner and fill the gap between business and technology by automatically transformingthe business model to working technology. It uses its MDA mark and these marks can be easily attached to the model and also detached whenever it is not required to the application. It prevents the model to pollute itself from the platform dependent specification details. ArcStyler uses the cartridges for code generation which provides flexibility. The cartridges are flexible MDA engine which generate code automatically. The cartridges contain all the details that are required to automate the model to a particular infrastructure.

Eclipse Modeling Framework (EMF)

Eclipse Modeling Framework is a Java/XML framework for creating tools and applications. Application created in EMF is based on class models. It helps to transform the models into java code in an efficient manner and at a very low cost. It also can save objects as XML and that XML documents can interchange from one application/tool to others. Models can be formed using Java and XML documents. The code generator transforms the model into a set of Java classes. The modification of the model can be done by either changes in Java code or by updating the model. EMF itself contains two models: one is the Ecore model that is the metamodel which describes the structure of the model and the second one is Genmodel that controls the generation of the EMF model [12][15].

Open MDX

OpenMDX is an open source MDA platform that helps the developer to build applications in an automated way based on Object Management Group. The main feature of open MDX is to develop MDA based application for java in which mapping from PIM to java is automatically handled and there is no need to do difficult work with platform specific applications [13].

Conclusion

Model Driven Architecture is an approach that provides means to fully utilize the advantages of system development using models. Portability, interoperability, reusability, and maintainability are improved when MDA is used. Even if there is a change in the software system that needs some newer technical stack, major changes would be in PSM only. Due to all these advantages, MDA has become a more commonly used platform and hence many tools are available in the market that implement MDA. Thus software development has become more standardized and organized.

References

[1] Y. Singh, M. Sood,“Models and Transformations in MDA”, First International Conference on Computational Intelligence, Communication Systems and Networks(2009).
[2] A. Aftab, M. Usman, Z.Halim“Model Transformation in Model Driven Architecture”, Universal Journal of Computer Science and Engineering Technology (2010).
[3] M. Lhioui,” A new method for interoperability between lexical resources using MDA approach”, Springer(2011).
[4] Y.Singh, M.Sood,”The Impact of the Computational Independent Model for Enterprise Information System Development “,International Journal of Computer Applications (2010).
[5] F. Abdelhedi, A. Brahim, F.Atigui, “MDA-Based Approach for NoSQL Databases Modelling”, Springer International Publishing (2017).
[6] H. Dev, A.Seth,” MDA based approach towards Design of Database for Banking System”, International Journal of Computer Applications (2012).
[7] Fabio Perez Marzulloet.al.,” An MDA Approach for Database Profiling and Performance Assessment”, Computer and Information Science, SCI(2008).
[8] I. Dubielewiczet.al., “An Approach to Evaluation of PSM/MDA Database Models in the Context of Transaction Performance”, IJCSNS International Journal of Computer Science and Network Security(2006)
[9] A. Rosa et.al. “An MDA Approach for Database Modeling”, Lecture Notes on Software Engineering (2013).

[10] S. Kumar Mishra et.al., "MDA based approach towards Design of Database for Online HealthCare System", International Journal of Computer Science and Information Technologies(2014).

[11] M. Polo, "An MDA-based approach for database re-engineering", Journal Of Software Maintenance And Evolution: Research And Practice (2008).

[12] Muliawan, Olaf," Reengineering JCMTG to MoTMoT: a migration from androMDA 2 to 3".Doctoral dissertation, Master's thesis, University of Antwerp, Belgium, (2005).

[13] W.Froidevaux. openMDX. http:/1/www.openmdx.org.

[14] http://andromda.sourceforge.net

[15] https://www.eclipse.org/modeling/emf

[16] C. Zhao, K. Zhang, "Transformational Approaches to Model Driven Architecture - A review",IEEE Software Engineering Workshop(SEW)(2007).

[17] Kumaran, Santhosh, et.ql. "Using a model-driven transformational approach and service-oriented architecture for service delivery management." IBM Systems Journal 46, (2007)

[18] Y. Singh, M. Sood," Model Driven Architecture: A Perspective", IEEE International Advance Computing Conference (IACC) (2009).

[19] T. Calic, S. Dascalu, et.al., "Tools for MDA Software Development: Evaluation Criteria and Set of Desirable Features", Fifth International Conference on Information Technology: New Generations(2008)

[20] Object Management Group, at http://www.omg.org