An Extensive Study on the Future of Modeling in Software Development

Avadhesh Kumar Gupta and Satish Kumar
Department of MCA
Institute of Management Studies
Ghaziabad, Uttar Pradesh, India
{dr.avadheshgupta, satishkumar.serg}@ gmail.com

Abstract--In recent years, software modeling realized much attention in the field of software research and development due to demonstrating the capability of decreases time and cost of software development and also improves overall quality of software development process in the form of understanding, communications, correctness, traceability, and developer productivity etc. Modeling has great potential to proven mainstream of software development, especially after using analysis and design phase to implement the system, because models are used as a primary artifact of software development and have capability to generate software implementation using model driven tools. This paper aims to describe the value of modeling in context of current mainstream of software development and also introduces different model driven approaches for software development. This paper also showed the growing acceptance of model driven technologies in software industries and current research initiatives of developing high quality model driven solutions for saving the future of modeling in software industries.

Index Terms: Modeling, Model Driven Engineering, Model Driven Technologies.

I. INTRODUCTION

Modeling is not a new paradigm in software development or any engineering discipline. For many years ago, mathematicians, engineers, scientist, system analyst and other technical professionals were used modeling for creating simplified and effective structure of complex system. The information we get about the real world problem is structured by the model, and also the whole thing is reduced to those phenomena and aspects which are considered essential [1].Model has great potential to explain phenomena, improve system predictions, decision making and effective communication and provides inside view of entire system before building and also make a direct cost effective benefit of building right system. Model can be used for obtaining better understanding of software system that helps software development team can get better communication with other users involved in the project and manage better complexity of software development process.

In earlier software engineering practices, models were used in different perspectives such as use case models can be used to define high level requirements of the software system, static models describes the static structure of the system being developed using UML class diagram, dynamic models provide better viewpoint of a system in the form of communication and sequence diagrams that provides better understanding of components interactions in run time environment [3,5].

Nowadays, Model Driven Engineering (MDE) is an emerging approach of software development; where models play a central role for software construction in the form of generating software implementation from these models. Domain Specific Language (DSL) is realized the actual value of models in software development [10]. DSL facilitates the software developer to create their own domain specific models and generate 100% code from these models using DSL tools such as MetaEdit+ [20]. Model Driven Architecture (MDA) is a standard framework proposed by Object Management Group (OMG) for software development, where models are designed by Unified Modeling Language (UML) and modeled application domain in different level of abstraction [OMG]. This paper is organized as follow. The next section describes the benefits of modeling in software industry. In Section 3, we have covered all cutting edge model driven approaches and discuss their technical advantages in software development. In section 4, we have studies the current place of model driven technologies in software industries. In final section 5 describes the future aspects of modeling in software industries.

II. BENEFITS OF MODELING IN SOFTWRE DEVELOPMENT

Nowadays, software systems are more and more complex due to need of integration with other software system to runs the system used in our daily life. For example, e-commerce applications are highly integrated with other web applications such as online banking payment system or other e-commerce applications, to develop such type of software system realize much attention towards security constraints and quality of services in the form of performance, interoperability and level of understanding [2]. For the reason, software engineers need to get better understanding about what they are going to develop.

Models are widely used in order to get better understanding of the software system by considering it from different perspective such as requirement models, conceptual models, specification models and implementation model of software system [3]. Unified Modeling Language (UML) is a general purpose graphical language that provides graphical notations for modeling real world object and gain much acceptance in the development of model driven engineering based software solutions. Modeling has numerous advantages in developing model driven solutions are as following [2, 4].

- Models are used as primary artifacts of software construction; in practice, these models are used for generating software intensive implementation which reflects in improving the productivity of software developers.
- Models are used to provide better understanding of the software system that helps to software developer create or develop high quality software system.
- Models facilitate early stage evolution of software system that helps to reduce the number of errors in the final software implementation.
- Verify the correctness of software system before the final software implementation using model verifications.
- Model provides better communication between development team and client
- Model provides an effective way of traceability throughout software development process.
- Modeling have the ability to manage better complexity of software development
- Model provides a better way of cost and time estimation.

III. MODEL DRIVEN APPROACHES FOR SOFTWARE DEVELOPMENT

Model Driven Engineering (MDE) is a promising approach to address software complexity and express domain concepts in effective way. MDE is also concerned with all aspects of reducing the technical gap between problem domain and software implementation domain by using modeling technologies that supports systematic way of transforming the problems domains into solutions domain [4, 5]. The primary intention of MDE is to raise the level of abstraction in software specification that reduces software complexity and increases automation in software development. This section cover all executable model driven approaches shown in figure-1 are described below

A. Model Driven Development

Model Driven Development (MDD) is one of the prominent approaches of software development in which models are considered as primary artifacts and these artifacts are used to generate the code and other sub-artifacts by using automation tools such as EMP, IBM Rational Software Architect etc [5].

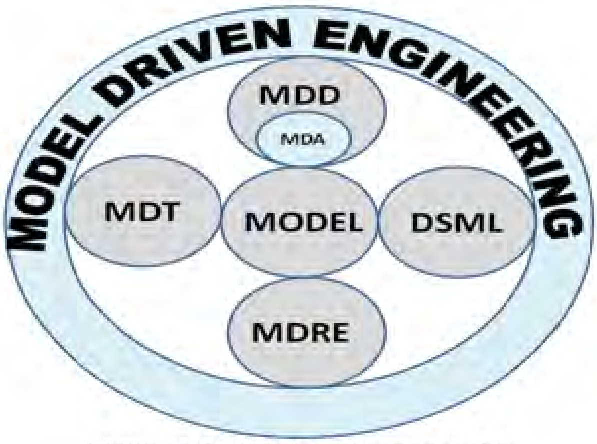

Figure:1 Model Driven Approaches

MDD has the potential to improve the quality of software in the form of following key aspects

- MDD improves productivity due to support of rapid automation in software development that decreases time and cost of software development by generating code and other sub-artifacts from the model and also increase the developer productivity.
- Maintainability is one of the big issue in Enterprise system, because technologies and platforms are changed frequently in the IT market and each organization want to shift their business system on new technologies or platform that create a problem of redevelopment of system that increase the unnecessary cost and Burden for the organization. MDD helps to solve this kind of problem by leading

to maintainable architecture where changes are made rapidly and consistently and enabling more efficient migration of legacy system on to new technologies or platform [6].

- MDD improve design communication, because models help in understanding and reasoning about system at design level and this leads to improve the discussion making and communication about the system.
- Model Driven Architecture is the particular vision of MDD proposed by Object Management Group (OMG) and widely accepted as a subset of MDD as shown in figure-1 [7, 8].

B. Domain Specific Modeling Languages

Domain Specific Modeling Language (DSML) is also known as Domain Specific Language (DSL). DSL has potentials to generate 100% code from domain specific models and increases software development, 10 times faster than traditional software development approach. In DSL, models are put at the core of software development, making it truly and completely model driven [9]. DSL has ability to deal with better complexity, raise the level of abstraction, making model easier and more convenient. As a domain specific modeling concept, it has limitation to develop software system only for a particular area in that company. But on other end, domain specific model constructions are better for communication with users in that particular domain and the users can get better understand the model, if they are presenting in the form of domain specific. It leads to claims that domain specific models are more productive for requirement engineering in comparison to other languages like UML [10, 11].

C. Model Driven Testing

Model Driven Testing (MDT) is an effective and well promising approach for software testing that supports rapid automation of test case generation from the domain specific UML models. MDT has the ability to reduce test maintenance cost, because at early stage of software modeling, the implementation changes are confined in the models [12,13]. MDT also facilitates to developers have customizable option to regenerate the test cases as per required changes affected to all test cases and also separate the testing logic from the test implementation. This permits the developers to concentrate on development application which is totally specific on test cases identified at the early stage of software modeling. Automated software test generation greatly decrease the cost and time of software testing using MDT tools [14]. These MDT tools facilitate the software tester to create and edit models in easy way, simulate models, generate correct and quality test cases from the models etc.

D. Model Driven Reverse Engineering

Today, software industries are facing lot of difficulties in managing software maintenance projects. Enterprise applications that have been developed with outdated technologies due to constant evolution of software technologies lead to continuous software modernization and migrations of software components on new platform or technologies [15]. These types of projects may be motivated due to enterprise centric software requirements. In traditional software engineering, software migration is accomplished through redevelopment of legacy system, while model driven software engineering provides better opportunity to increase automation in software migration. Here, Model Driven Reverse Engineering (MDRE) play a key role in obtains high level of abstraction of legacy system and to create models that represents the system at different level of abstraction [16]. These models can also be used for different purpose such as documentation generation, metrics and quality assurance computations and software evolution and modernization process etc [17, 18]. By applying MDRE in legacy software migration or software modernization, ensures the high automation of software migration or modernization, but have difficulty to achieve full automation due to the technological gap between the legacy platform and the new platform. MDRE ensure the quality of new system and facilitates the developer to create new version of the application using advanced development techniques.

IV. MDE ADOPTION IN SOFTWARE INDUSTRIES

Today, model driven approaches provide industry leadership that is resulting in significant technological advancements and customer satisfaction. We have done extensive survey of acceptance of model driven technologies in several software industries and we found, the growing acceptance of model driven approaches in software industries. Several research & development projects are running in model driven area that shows much interest of software industries. Eclipse Foundation is one of core Software Company which provides Open Source MDE based Tools for software development. This organization is running several R & D project like Eclipse Modeling Project, Graphical Modeling Framework and Epsilon etc for improve the overall quality of development [19]. MetaCase is one of the core model based software company that provides MDE/DSM based software solutions in various domains like Telecommunications, Embedded, enterprise applications, mobile applications, and automotive industries etc. In MetaCase Technology, software solutions have developed using domain specific modeling languages with help of MetaEdit+ tool [20]. MetaCase tool facilitates the developer or domain modeling professionals to create one or more modeling languages, methods and notations for use within the process of software development. MetaEdit+ tool has the potential to develop high quality software and supports 100% code generation from the domain specific models [21]. In MODELIOSOFT, everything is transform in the form of model driven technologies. It shows rich adoption of modeling technologies in various domains such as enterprise application development, enterprise architecture, business architecture, BPMN, SOA, model driven support of an open source system development [22]. ADA Software is strongly involved in model driven cutting edge solutions and IT services in various industrial domains such as MDA & IT strategy consulting-commerce application, Healthcare, Digitalization, enterprise solutions, embedded system and specialized resource in model driven legacy modernization and component software factory [23]. Model Driven Solutions is leading provider of MED based IT solutions in various domains such as enterprise architecture solutions, model driven information sharing with NIEM-UML, model driven architecture based solutions for business centric application etc. MDS is actively involved in several MDE based R & D projects such as ModelPro, Foundational UML etc [24]. John Hutchinson [25] has done empirical study of MDE adoption in different industries such as Printer Company, Car Company, and Telecom Company and found great acceptance of model driven engineering in different aspect of product design and development. Some of them are described as following

- In printer company case study, MDE is used to overcome the problems that could not be solved using existing development technologies. Applying MDE in product development, it improves the overall quality of production and decrease complexity of product development in the form of technical aspects and time [25].
- In Car Company case study, MED is not adopted due to technical reason but also it required a solutions of commercial and organizational challenges that could not be solved and have difficulty to using old technological infrastructure [25].

Kulkarni at TCS [26] provided practitioner and industry look into future of MDE based software development. Managers are agreed to the qualitative benefits of MDD and transforming their core software production on MDE based technologies.

V. FUTURE OF MODELING TECHNOLOGIES

After long study of model driven technologies, the growing acceptance of model driven engineering in software industry showed much interest. Top MDE based solutions providers are actively involved in several research and development projects for developing high quality open source MDE based software packages and tools. These tools accelerate software development with promising aspects in the form of achieving better software quality, maintenance, and production etc. It given a very clear and straight forward message; in future, model driven technology is one of the core drivers in software industries.

VI. CONCLUSION

Software modeling has proved its value in model driven development. Modeling recognizes as a mechanism to obtain better understanding of software system that helps software developers to manage better complexity of software development and also supports in rapid automation of generating software artifacts from domain specific models. The growing adoption of model driven technologies in software industries showed the future of modeling and various open source communities are doing cutting edge research in developing model driven open source packages for improve the overall quality of software development. Finally, extensive study of software modeling has shown the present and future of modeling in software industries.

VII. REFERENCES

[1] H. Schichi, “Models and History of Modeling”
URL: http://radon.mat.univie.ac.at/~herman/papers/modtheoc.pdf

[2] G.Cemosem and E.Naiburg, “The Value of Modelling” A Technical Discussion of Software Modeling, June 2004, URL: http://www.cs.umb.edu/~jxs/courses/2007/680/papers/ValueOfModeling.pdf

[3] H. Gomaa, “Software Modeling & Design” Cambridge University Press, New York, Edt. 2011.

[4] M.Staron, “Adopting Model Driven Software Development in Industry-A Case Study at Two Companies”, MODELS 2006, LNCS 4199, pp. 57-72 , Springer-Verlag Berlin Heidelberg, 2006.

[5] D.C. Schmidt, “Model Driven Engineering” IEEE Computer Society, pp. 25-31, 2006

[6] B. Nolan, B. Brown, T. Bohn and L. Balmelli, “Model Driven Systems Development with Rational Products”, IBM Rational System Developer, Version 7, First Edition, 2008.

[7] R. Soley, A.Watson, C. Burt, and B. Wood, “Model Driven Architecture (MDA) Guide Version 1”, Object Management Group, July 2001., URL: http://www.omg.org/cgi-bin/doc?omg/03-06-01

[8] K. Goede and J. Irizarry, “Understanding Tools Requirements for Model Driven Architecture”,OMG, 2008, URL: http://www.omg.org/mda/mda_files/Understanding_MDA_Tool_Requirements2.pdf

[9] J. Sprinkle, B. Rumpe and G. Karsai, “Metamodelling: State of the Art and Research Challenges”, MBEERTS, LNCS 6100, pp. 57-76, Springer-Verlag Berlin Heidelberg, 2010.

[10] J.P. Tolvanen, “Domain Specific Modeling: How to Start Defining Your Own Language”, 2006, URL:
http://www.devx.com/enterprise/Article/30550

[11] J. S. Cuadrado and J.G. Molina, “Building Domain-Specific Language for Model Driven Development”, IEEE Software, pp. 2-8, 2007.

[12] M. Mussa, S. Ouchani, and W.A. Sammane, "A Survey of Model-Driven Testing Techniques", International Conference of Knowledge Engineering and Software Engineering, pp. 55-62, Canada, 2005.

[13] R. Heckek and M. Lohmann, "Towads Model Driven Testing", NTCS 82, Elsevier Science, 2003.

[14] IBM Haifa Research Laboratory, "Model Driven Testing Tools", 2003, URL: https://www.research.ibm.com/haifa/projects/verification/mdt/papers/MDTWhitePaper.pdf

[15] J.M. Favre, "Foundation of Model Driven Reverse Engineering", Post-Proceedings of Dagsthul Seminar on Model Driven Reverse Engineering, 2004.

[16] F. Fleurey, E. Breton, B. Baudry and A. Nicolas, "Model-Driven Engineering for Software Migration in Large Industrial Context", MODELS 2007, LNCS 4735, pp. 482-497, Springer-Verlag Berlin Heidelberg, 2007.

[17] S. Rugaber and K. Stirewalt, "Model-Driven Reverse Engineering", IEEE Sosftware, Vol. 21, pp. 45-53, 2004.

[18] M. Brambilla, J. Cabot, and M. Wimmer, "Model Driven Software Engineering in Practice", SLSE, Morgan & Claypool Publisher, 2012.

[19] Eclipse Foundation, "Eclipse Modeling Framework Project & Graphical Modeling Project", URL: http://www.eclipse.org/modeling/

[20] MetaCase Technology, "Domain-Specific Modeling with MetaEdit+", URL: http://www.metacase.com/cases/dsm_examples.html

[21] MetaCase Technology, "Domain-Specific Modeling with MetaEdit+: 10 Time Faster then UML", URL: https://www.metacase.com/papers/Domain-specific_modeling_10X_faster_than_UML.pdf

[22] ModelioSoft, "Model Driven Architecture Technology & Comprehensive Diagram Support with Modelio", URL: http://www.modeliosoft.com/en/resources/diagram-examples.html

[23] ADA Software, "MDA FastStart", URL: http://www.adasoftware.com/domains_mdafaststart.htm

[24] Model Driven Solution, "Model Driven Architecture", URL: http://www.modeldriven.com/MDA.shtml

[25] J. Hutchinson, M. Rouncefield, and J. Whittle, "Model-Driven Engineering Practices in Industry", ACM International Conference on Software Engineering, pp. 633-642, 2011

[26] V. Kulkarni, "Model Driven Software Development: A Practitioner Takes Stock and Looks into Future", ECMFA 2013, LNCS 7949, pp. 220-235, Springer-Verlag Berlin Heidelberg, 201

AUTHORS PROFILE

Dr. Avadhesh Kumar ***Gupta*** has more than twelve years of teaching and research experience in the field of Computer Science. He holds M.C.A., M.Tech. in Information Technology, PhD in Big Data Computing. He is a Professor and Head of MCA Department at Institute of Management Studies Ghaziabad (IMS-GHAZIABAD), Uttar Pradesh, India. He has several national and international research publications to his credit. He is an active member of ACM Computer Science Teaching Association USA, IAENG Hong Kong, IACSIT Singapore and IEEE Computer Society.

Mr. Satish Kumar is a teacher and professional trainer in the field of developing ICT based Infrastructure. Currently working as a faculty in MCA Department at Institute of Management Studies Ghaziabad (IMS Ghaziabad), Uttar Pradesh. He received Master of Technology (M.Tech.) in Information and Communication Technology. His research interests in engineering of self-adaptive systems, particularly in developing software intensive distributed systems. He is life time member of ACM computer science teacher association USA and IAENG Hong Kong etc.