International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248
Volume: 3 Issue: 12 46 – 50

Model Driven Architecture: A Review of Current Literature

Dr. Harsh Dev
Professor
Department of Computer Science and Engineering,
PSIT,
Kanpur, India
E-mail: *drharshdev@gmail.com*

Swasti Agrawal
Research Scholar
Department of Computer Science,
IFTM University,
Moradabad, India
E-mail: *swasti.kanha@gmail.com*

Abstract: There are numerous Model Driver Engineering (MDE) methodologies but Object Management Group (OMG) approved of Model Driver Architecture (MDA). MDA methodology has a target to systemize the software progressing procedure with the use of models rather than the old-fashioned coding based on isolation of the related theory. During the month of June in the year 2014, OMG brought second edition of MDA guide into the market in attempt to understand about essential values and to back first edition of MDA guide which came out in 2003 and had thorough provisions included within. An interval of 11 years allows the investigators to come out of behind and put forward their viewpoint with the various clarifications of MDA provisions. People often gets mistaken and consumed about what is outside of MDA scope and what is inside it. Severely mentioning to MDA standard (not MDE in general), a review of present MDA Literature is given by us here. A bit of a spotlight is also cast upon the MDA research directions, more particularly upon mechanizations of MDA progress procedure and the raised areas which it aims.

***Keywords**: MDA, MDE, OMG, Model Driven Architecture*

*****

I. INTRODUCTION

Once it gained attainment in giving a technology autonomous groundwork regulation (CORBA) [1], OMG quickly shifted from the past Object Management Architecture (OMA) image [2] with an acceptance of Model-driven Architecture (MDA) methodology. OMG needed to inspire MDA methodology therefore the OMG assisted MDA with the uniform provisions for Unified Modeling Language (UML) [4], Meta-Object Facility (MOF) [5], XML Metadata Interchange (XMI) [6] and Common Warehouse Meta-models (CWM) [7]. Such provisions are the representative of a central groundwork of MDA [8]. It aids in understanding about the isolation of the related theory too and thus grows the extent of corporation and interoperability among the coordination.

Very first edition of MDA values was publicized by OMG [9]. That edition comprised of complete description for MDA provisions and was looked at as practical orientation for MDA practitioners. The latest edition by the OMG came out in the month of June in year 2014 which had comparatively lighter information than the edition before it [10]. The single drive of it was to form a business case for MDA. OMG's quietness for the eleven long years forced investigators as per their upbringing to come up with fresh provisions and theories from other MDA methodologies beneath MDA framework. Therefore, new MDA acceptors consider such provisions and theories as actual MDA values.

Here, we would restrict the limitations of genuine MDA theories. MDA was also ranked in between other traditional MDE and old-fashioned software progress methodologies. It is an analysis of present MDA literature. It does not comprise of a literature for the other model driven methodology.

Primary aim of the research is that it gives an important analysis for genuine MDA values which are stated by OMG. The studies or the software developers who chose the MDA, this study would give them the path for the ride.

Subsequent segment of study gives an analysis of present MDA literature and study guidelines with an attention towards MDA progress procedure and aimed areas. Procedures of analysis are shown in Section and details in Section 4. At last, we can finish with outcomes, suggestions and the work for time ahead in the section 5.

II. LITREATURE REVIEW

A. Model Driven Architecture

Essential aim of the MDA is to originate the importance from the models which back aid us to compress the complications and the interdependence of the compound structures. Isolation of related theory attained by the three architectural layers: At the first layer, computational independent model (CIM) seized business and domain vocabulary which was frequently given by the business specialists. The layer fills the interval between field specialist and executors of schemes. Second layer is Platform Independent Model (PIM) and it validates the data in CIM model autonomously from any technology platform. Platform Specific Model (PSM) acts as third layer which has an attention on technical and platform execution information.

Regardless, even now the PSM is looked at as theoretical to be implemented on a computer field. Therefore, Platform Model (PM) is rearranged to aid PSM and serve as a mechanical guide for aimed field. PM indirectly occurs in the model conversion principles. Indirect application of PM through the course of conversion is restricted model conversion to talk about a solitary assumed field [11]. Something which forms the sureness regarding the probability of employing the model plotting for other fields different than one for which it was intended. Conversion or plotting of model is primary doing in MDA. Considering the Platform Model (PM), it acts in converting the high-level models to low-level models and the other way around. Also, conversion could be inside same level of generalization (PIM-PIM) or (PSM-PSM). Such situations of plotting are categorized as Model-to-Model (M2M) plotting or Model-to-Text (M2T). Both of them are beneath MDA and backed by numerous instruments which fitted out to talk about every situation and the plotting [13]. Conversion procedure mechanization among MDA models is distinguished from other MDAs. This is because the methodologies like MDE, Model-Based Engineering, and Model-Driven Development (MDD) and the models which originated as communication methods among system investigators or to openly produce the code from the models. Avoiding architectural layers of MDA and absence of mechanization in progress procedure, Figure 1 ranks MDA in between the other model-driven based methodologies [13].

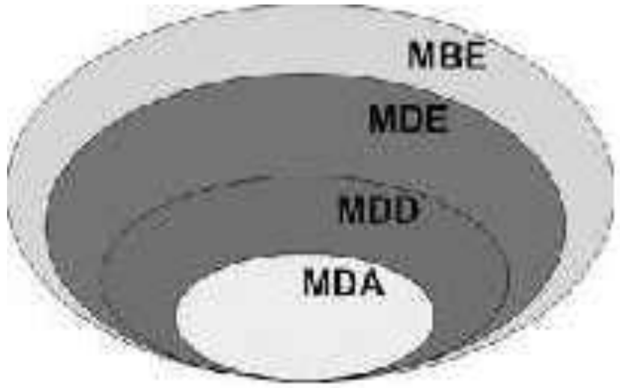

Figure 1. Positioning The MDA Among Other

Mechanization too provides MDA with a superior advantage in software efficiency and cost lessening. The Figure 2 (a) signifies traditional software progress procedure. Just about any probable mechanization occurs at coding level below. Therefore, alterations in coding do not have any effect on upper levels while the changes on top levels above coding are looked as a substantial attempt in recoding. It is due to text and diagrams in above layers being frequently made use of for the certification and communication intentions [14]. MDA procedure is displayed in Figure 2 (b) ad it shows a computerized locked circle of the levels where alterations in theoretical top levels promulgated by themselves to lower ones with the less attempt, time and cost. Conversely, alterations in lower theoretical levels are seen in upper levels if we move back the conversion from lower level code to upper layers with the higher level of abstraction models [15].automation as well gives the MDA an upper hand in software productivity and cost reduction. Figure 2 (a) represent the classical software development process. Practically, any possible automation is taking place at the coding level downward. Consequently, changes in coding don't reflect the top levels, and updates on the top levels above the coding on the other hand means a considerable effort of recoding. This is because the text and diagrams in the above layers are commonly used for documentation and communication purposes [14]. The MDA process is shown in Figure 2 (b), present an automated closed loop of between levels. Where, changes in the abstract top levels propagated automatically to the lower ones, with minimal effort, time, and cost. On the other hand, changes in the lower abstract levels can be reflected in the upper ones, by reversing the transformation from lower level of abstractions code/artifacts to the upper layers with higher level of abstraction models [15].

B. Issues Facing MDA

Restrictions made in aforementioned segment about what is MDA and rank of MDA in between other model-driven methodologies, its power is also seen when compared to traditional software progress and the other model-driven methodologies. It is clear now regarding the things inside the MDA scope and outside them. Here, the hurdles and the problems are demonstrated too which arrive while coming face to face with MDA by the investigators.

Prior to legit acceptance by OMG of MDA, work in [16] shows aim and benefit of MDA but it also noted technology variance and dynamicity as an essential problem which complexes the system incorporation and interoperability. With all this going on, software progress methodologies and lifecycle comprising of MDA, thing which forms a desire to control an amount of abstraction of MDA models if models are to be implemented at aimed field. Therefore, series of model conversion applied to talk about the platform atmosphere which means, a group of latest PSM models. Handling the latest PSM models with the related conversion regulations and methods is quite a difficult thing to do. It cannot be assured that the enterprises be inclined to a single middleware such as CORBA due to the corporation and latest attainment of the enterprises.

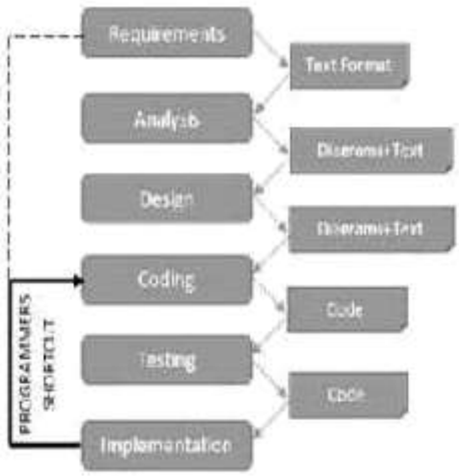

Figure 2(a): Classical Software Process

Conversion of the model is a primary task in MDA. Various approaches and instruments exists which can talk about the model conversion. A study held by Van et al (2004) had a central focus on conversion models and the instruments which assisted the model to model conversion whereas [9] is inspiring Model to Text Conversions. As they provide the evaluation of MOFscript language which was proposed to OMG as an initiated model to text conversion language. Presence of various conversion approaches backed by few conversion instruments in market make it uneasy for the MDA practitioners to select the instruments which can assist which conversion approach.

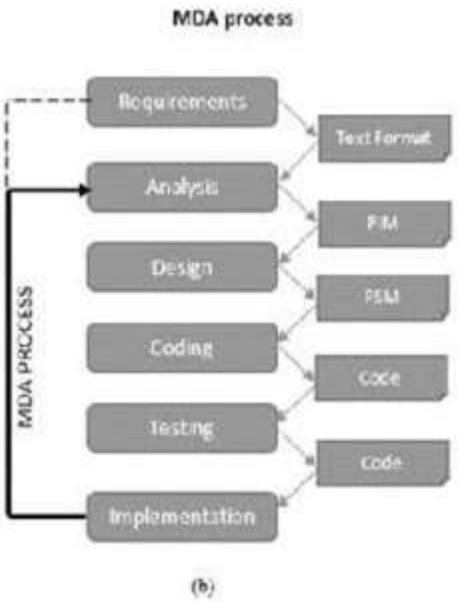

Figure 2 (b): MDA Lifecycle

Conversely, a dissimilar level of abstraction in MDA results in a series of model conversions to go from high level to lower level abstractions. No assurance can be made that the model conversion could be used for any other field other than the one for which it was intended. PSM aims a specific field. Though, even now it is looked as too theoretical to implement on anything. Additionally, every platform varies. No platform is same with any other. An appropriate agenda is needed which has the ability to carry the conversion with lesser alterations and specific information regarding the implementation atmosphere. Such problems are addressed by [19] as they made use of the ontology to describe the components of fields. Though, mechanization of managing remarkable ontology is not possible because of numerous groups and happenings. Besides, time spent in development of ontologies develops to be even complicated on variety and information scale of the platform.

There is one more problem and that is MDA is totally related to mechanization of progressive lifecycle and software efficiency. Developers require mechanized methods which are extensible, universal and accessible. Though, MDA procedure has an absence of fitting instruments which can assist the mechanization. Work in [20] initiates an agenda to back the MDA mechanization guidelines. Adjacent to demonstration to platform element, point of reference to instruments is given which assists the MDA mechanization goal.

C. MDA in Action

Considering the aforementioned problems, latest guidelines inspire for sidestepping a few of MDA layers or addition of fresh layers. Work in [19] in inspiring to remove PSM and talk about aimed field openly from PIM. It is alike to the supple approach which [20] has made use of. Other supplest such as [21] are racing to catch a vision of possessing an implementation model without the thought of MDA values and by openly reaching towards UML 2.0 profiles which give an extent of suppleness in stating regarding the components of the field. Supple MDA is a prevailing theory which is more appealing for the software organizations as the part of the model moves from certification to the carrying out. In spite of the assurances, MDA was labelled as a long-lasting procedure which can result in providing incorrect system late at higher expenditure. The supple approach inspires for implying the theory of that code and model are operationally similar where implementation model can iteratively be developed, used, examined and altered in the short cycles [21].

Conversely, methodologies such as ArchMDE [22] were seen to be attempting to form a new standard which permits a formulation of software architecture from evaluation model. While they try to initiate an addition of a new layer (AIM) in MDA theory, remaining investigators are recommending of engaging CASE instruments as a various direction which is combined in between MDD and MDA [23]. Traditional CASE instruments structured on a basis of classic database theory where the models are kept in a retrievable source. Work in permits us to have suppleness of signifying models in a written set-up which can be converted in XML set-up. Written set-up provides more control in model restraints and reliance control and is assisted by MDA provisions.

 ISSN: 2454-4248
46 – 50

Alike to that, work in gives an End to End conversion agenda from PIM to PSM inside MDA extent to assist software product line. Additionally, work in [26-28] showed a high extent of obligation to MDA values.

III. REVIEW PROCES

Review process trailed here is on the basis of glancing over OMG MDA database, and the provision's segment. MDA along with remaining model-drive methodologies in various academic databases such as Google Scholar, Springer's, IEEE Explorer and ACM were glanced over too. A functional description procedure is to be gone through to describe what is inside the MDA extent by mentioning the [10] as an establishment to compare MDA methodology with remaining model-driven methodologies on the basis of these 4 measures:

- Acceptance of isolation of related theory.
- MDA architectural layers conformation.
- Conversion approach
- Managing Platform Variety against mechanization of progress lifecycle.

Figure 3 demonstrates high level of MDA theory as per OMG principles [12]. Distress is split to business fields which comprise of CIM & PIM whereas PSM and PM are platform and technology attentive. Conversion fields divided to conversion provisions and conversion instruments. Both of them shall assist the management of abstraction scale of the model.

Figure 3: MDA High Level Architecture

Study in this work which expresses a theory of altering the present MDA architectural layers lectured by analyzing an applicable kind of subject and studies in model based software progress procedure field. Assessment guidelines for analyzed literature are on the basis of high-level architecture in Figure 3 with the additional two MDA guides which the OMG publicized [12].

IV. DISCUSSION

As MDA is targeting to project and construct software product which can be employed in various platforms with slight manufacturing by making use of the models, there are other methodologies such as MDE and MDD which focus on the model. Such methodologies have a target to improve the standard of software, efficiency, interoperability, cost and time to market. Though, both of the methodologies function on dissimilar levels to attain a similar target. MDA is making use of the models in higher isolated concerned abstraction levels to generate software neglecting hard coding in end to end mechanization whereas other model driven methodologies do not give attention on procedure mechanization or isolation of concern. However, models are made use of in the future for the purpose of certification or communication.

Lack of OMG provisions in understanding MDA theories forced the investigators to step up and provide own creativeness from other model driven methodologies on the basis of their potential and contextual. Therefore, a misperception controls results of the study and MDA basic rules had to adjust.

Supple MDA is an attention-grabbing field which we talked about earlier particularly the part of model combination. PSM can be combined with PM as both of them are models and can signify in a written or visual way. Removal of PSM or addition of a new layer to MDA procedure has an absence of an instrument to assist. MDA quality instruments do not have the means to manage these steps. QVT which focuses on model-to-model conversion to possess the implementation code we require to make use of MOF model to text which therefore, would be restricted to a limited number of languages like EMF or ATL.

V. OPEN RESEARCH ISSUES IDENTIFIED

Various guidelines in this study are given which are separated in two: first guideline concentrating on MDA set-up and instruments which assist the values. Investigators making use of this guideline possess a restricted area to work if they want to incline on MDA values and attain the acceptance of OMG for the methodologies and instruments. It is due to the efficiency of OMG in giving MDA adopters with improved provisions. Therefore, various instruments exist which can accomplish numerous MDA basics [15].

Model-to-Text is a decent instance for such circumstance. We have quality but instruments and studies which are requires for these provisions are too much restricted and therefore, new MDA joiner is at the end of its tether for the instruments which can give a good level of certification and assistance.

Second study direction which focuses on MDA presentations. This direction develops rapidly and reaches to a decent scale of prime of life. Various MDA applications and studies in the field of cloud work out, software analysis and rooted system have a remarkable outcomes and accomplishments [17].

VI. CONCLUSION AND FUTURE WORK

Computer fields are merging few factors and the elements which permit to legally manage various operations and backgrounds which are typically restricted to a specific field of technology. With the ability of generating software from models to back more than one platform for various technologies at various levels of abstraction is intensely elevating the standard and efficiency of software progress procedure. MDA standards can be involved affirmatively in software standard and efficiency and especially how dissimilar it is from the remaining model oriented methodologies. Significance of the study is that it can be made use of to comprehend the existing rank of MDA and how to shadow the guidelines. Additionally, for such investigators who are willing to imply other methodologies can use the study to preserve the status of their work. Problems regarding to the MDA development are assessed and displayed in the study. The problems are related to all model driven methodologies in one way or the other.

REFERENCES

[1]. Selic, B. (2003) the Pragmatics of Model-Driven Development. IEEE Computer Society. IEEE Software, Pp. 19n25.

[2]. Selic, B. (2004) Model-Driven Development in the Embedded Environment with OMG Standards. Presentation in the second international summer school on MDA for embedded systems, Brest, Brittany in France.

[3]. Miller, J. & Mukerji, J. (2003) MDA Guide Version 1.0.1. Object Management Group. 62 p.

[4]. Ramljak, D., Puksec, J., Huljenic, D., Koncar, M. & Simic, D. (2003) Building enterprise information system using model driven architecture on J2EE platform. In: Proceedings of the 7th International Conference on Telecommunications, IEEE. Pp. 521ñ526.

[5]. Matinlassi, M. (2004) Quality-driven Architecture Model Transformation for the Software Product Families. Submitted to the Journal of Software and Systems Modelling. 32 p.

[6]. Merilinna, J. & Matinlassi, M. (2004) Evaluation of UML Tools for Model Driven Architecture. In: 11th Nordic Workshop on Programming and Software Development Tools and Techniques, Turku, Finland: Åbo Akademi. Pp. 155ñ163.

[7]. Bass, L., Clements, P. & Kazman, R. (1998) Software Architecture in Practice. Reading, Massachusetts: Addison-Wesley. 452 p.

[8]. Bosch, J. (2000) Design and use of software architectures: adopting and evolving a product-line approach. Harlow: Addison-Wesley. 354 p.

[9]. Van der Linden, F., Bosch, J., Kamsties & Obbink, H. (2004) Software Product Family Evaluation. In: Proceedings of the Third International Conference on Software Product Lines, Springer Verlag: Boston. Pp. 110ñ129.

[10]. Matinlassi, M. & Niemel, E. (2003) The Impact of Maintainability on Component-based Software Systems. In: 29th Euromicro Conference (EUROMICROí03), Turkey. Pp. 25ñ32. 105

[11]. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M. (1996) Pattern-oriented software architecture ñ a system of patterns. Chichester, New York: Wiley. 457 p.

[12]. Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1994) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series, Addison Wesley. 416 p.

[13]. Matinlassi, M., Nieme, E. & Dobrica, L. (2002) Quality-driven architecture design and analysis method. A revolutionary initiation approach to product line architecture. Espoo: VTT Technical Research Centre of Finland, VTT Publications 456. 128 p.

[14]. Nieme, E., Kalaoja, J. & Lago, P. (2004) towards an Architectural Knowledge Base for Wireless Service Engineering. IEEE Transactions on Software Engineering, vol. 31. 46 p.

[15]. Erikson, H., Penker, M., Lyons, B. & Fado, D. (2004) UML 2 Toolkit. Wiley Publishing Inc, Indianapolis, Indiana. 511 p.

[16]. Berkenktter, K. (2003) Using UML 2.0 in Real-Time Development: A Critical Review. In: SVERTS: Specification and Validation of UML models for Real Time and Embedded Systems, October 20. 14 p.

[17]. IEEE (2000) Recommended Practice for Architectural Descriptions of Software Intensive Systems in Std-1417-2000. New York: Institute of Electrical and Electronics Engineers Inc. 23 p.

[18]. Object Management Group (2003). UML 2.0 Superstructure Specification, 8.9. 623 p.

[19]. Bjkander, M. & Kobryn, C. (2003) Architecting Systems with UML 2.0. In: IEEE Computer Society, July/August. 5 p.

[20]. Gardner, T., Griffin, C., Koehler, J. & Hauser, R. (2003) A review of OMG MOG 2.0 QVT Submissions and Recommendations towards final standard. In: 1st International Workshop on Met modeling for MDA, York. 20 p.

[21]. Burt, C., Bryant, B., Raje, R. & Auguston, M. (2002) Quality of service issues related to transforming platform independent models to platform specific models. In: Proceedings of the Sixth International Conference on Enterprise Distributed Object Computing. Pp. 212ñ223.

[22]. Frankel, D. (2003) Model-Driven Architecture, Applying MDA to Enterprise Computing. Indianapolis, Indiana: Wiley Publishing Inc. 328 p.

[23]. Christoph, A. (2004) Describing Horizontal Model Transformations with Graph Rewriting Rules. In: Proceedings of Model-Driven Architecture Foundations and Applications. Pp. 76n91.