

# Model Driven Architecture - Issues, Challenges and Future Directions

Amna Noureen\*, Anam Amjad, Farooque Azam

Department of Computer Engineering, College of EME, National University of Sciences and Technology (NUST), H-12, Islamabad, Pakistan.

\* Corresponding author. Tel: +923365119708; email: Amna\_noureen@yahoo.com

Manuscript submitted December 31, 2015; accepted July 22, 2016

doi: 10.17706/jsw.11.9.924-933

**Abstract:** Model driven architecture provides the models written in well-defined language. They consist of multiple components including model, transformation and meta-model. This architecture is improvement of software development life cycle but it also comprises of several issues, challenges which needs to be addressed. Our focus in this paper is to elaborate the issues and challenges of MDA. Furthermore, we will propose some future directions.

**Key words:** Model driven architecture (MDA), meta object facility(MOF), object management group(OMG), unified modeling language(UML).

## 1. Introduction

OMG proposed model driven architecture for the large software system. It consists of model transformations and model refinements. MDA are carried out until system is executed and produced having abstract models which are the result of transformation of concrete models by adding details of technical sort. In Fig. 1 it is also mentioned that within the OMG, the MOF is the standard M3 language. All modelling languages (like UML, CWM) are used as instances of the MOF.

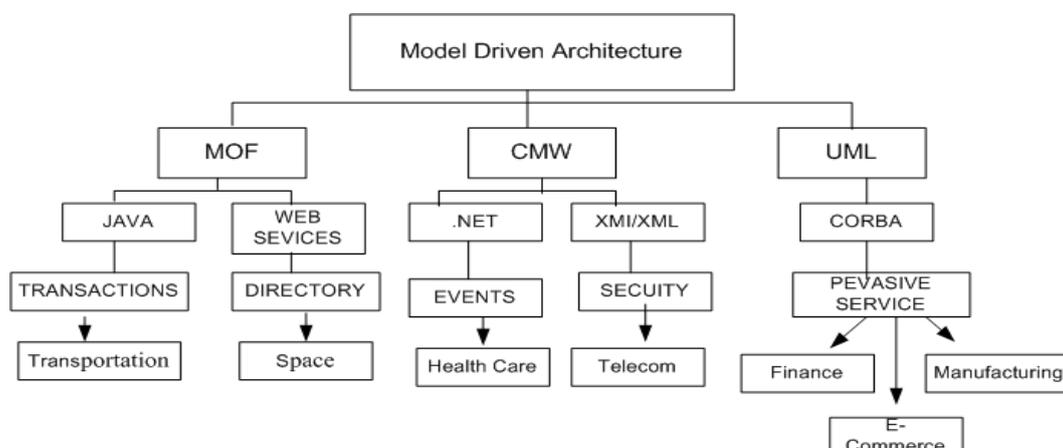


Fig. 1. Model driven architecture.

### 1.1. Components of MDA

The components involved in MDA are as following.

### **1.1.1 Model**

Models are used to build the large-scale solutions. Notations are used to represent the models. Four types of models are introduced in model driven architecture named as: Computational Independent Model (CIM), in which business model does not depend on the system model and implementation details are not present in it. Platform Independent Model (PIM), the name indicates that it is independent of any platform or operating system detail. Platform Specific Model (PSM), the PIM can be transformed in multiple PSM which are dependent on a particular platform or operating system. Implementation Specific Model (ISM), in which all details of implementation are specified [1].

### **1.1.2 Model transformation**

Different models can be transformed through some sequential steps in order to develop the system [1].

### **1.1.3 Meta-model**

Models are written in a well-defined language which comprises of another model used to integrate and transform the model called Meta models. Tools are used for this purpose OMG is using four layer of architecture known as M0, M1, M2 and M3 [1].

In this paper, we have described some issues and challenges of MDA. MDA is widely used methodology thus problem regarding to it should be processed and solved for its effective use and reliability.

The paper is organized as follow, section 2 provides the brief analysis of MDA which are further categorized into issues and challenges, section 3 contains the literature review of several papers related to MDA, section 4 we have presented the proposed solution and show how this proposed solution can be augmented with this model driven architecture technology. In last, section 5 provides the conclusion along with opportunity to enhance the work.

## **2. Analysis**

MDA is emerging field in software engineering. It has several advantages which are

- It provides long term flexibility through PIM
- New PSM are developed and updated
- Complex system can be views from different perspectives by using MDA
- MDA is used to keep the investment of both types; intellectual and financial which is very helpful for the developer
- MDA supports the application at multi-platforms
- Integration of applications is easier with MDA. It also facilitates the middleware boundary problems.
- Interoperability between two or more platforms is also present in MDA. The mean reason behind is the construction of PIM which is transformed in PSM.
- With MDA, cost of developing a software application is reduced as compare to the cost of developing software by traditional SDLC.
- Quality is not compromised in MDA, as it is clear that code is generated from the models created.
- MDA ensures that if technology framework is changed then existing applications do not become useless.
- MDA is wide scope methodology and due to its benefits to the developer it will be used in future too. .

Despite of all these advantages, MDA is also facing several issues and challenges. Some of the issues and challenges are discussed.

### **2.1. Issues of MDA**

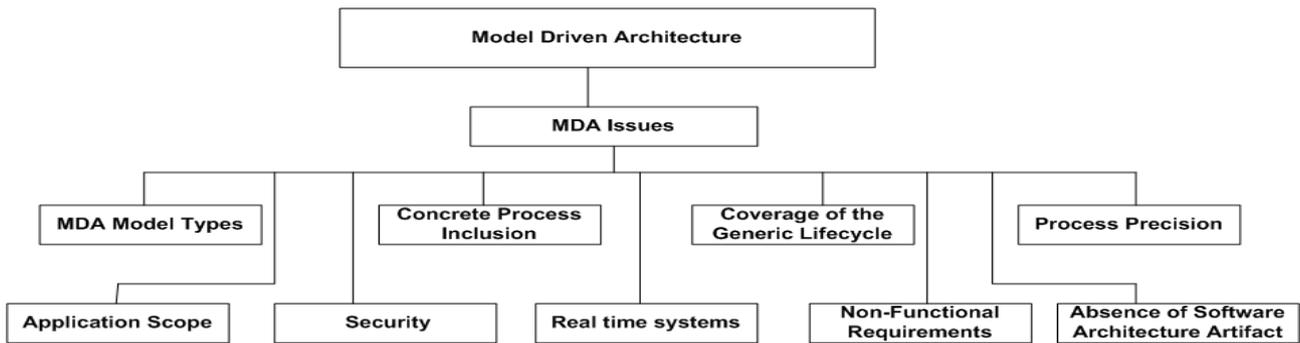


Fig. 2. MDA issues.

In Fig. 2 some common issues regarding MDA are highlighted and these are explained as follow.

### 2.1.1. MDA model types

Different methodologies of MDA support different idea, doctrines and methods. In this instance, C3 uses MDA standards which include UML and XMI then transformation is performed in the planned process, but model types are not integrated, i.e. PIM, PSM [1].

### 2.1.2. Concrete process inclusion

Various methodologies claim to be MDA-based do not proceed towards concrete process. Although they satisfy the process needs. [1]

### 2.1.3. Coverage of the generic lifecycle

In coverage of the generic life cycle, the whole issue resides to what extend phases and activities of the general SDLC is cover by the methodology [1].

### 2.1.4. Process precision

While developing the software, it should define the process efficiently and effectively by the experts involved in the development process Model driven architecture methodologies are responsible to describe the process but not in an effective and efficient manner. For example, MIDAS and MASTER do not fit into the behavior of their processes [1].

### 2.1.5. Application scope

Some MDA methodologies are not general-purpose and they are not applicable to the all domain. For example, MIDAS is application of web information systems and MODA-TEL focus on distributed applications [1].

### 2.1.6. Security

A software product can be secured enough if this factor is considered from the very beginning. When security is handled then security protocols scheme are occurred at minimum level. A MDA (model-driven approach) for the development of security vital applications is helpful in producing an execution and official model, from the identical PIM. They are required to guarantee correctness and security. [2]

### 2.1.7. Real time systems

Developing real-time system based on MDA is also significant. Some efforts have been made but it is partial, it can solve the rambling difficulty associated with function of RTS, this is not available in traditional SDLC methodologies. [3]

### 2.1.8. Non-functional requirements

There is variety of issues in satisfying the non-functional requirements in MDA such as modeling of non-functional requirements and their relationships, level of satisfaction of NFRs at run time. To deal with such issues, system needs to be monitored during runtime and if there is any violation, system requires reconfiguration by introducing new functional properties. The example of this scenario can be seen at

security level where security calls for fingerprint or multilevel authentication. In this case multi-level authentication might conflict with usability requirement in terms of time consuming process to attain security. An adaptive approach is needed for usability requirements at runtime by keeping a record of timing behaviors of encryption algorithms used for multilevel authentication and using a stronger one-level authentication with less time consumed when timing violation are supposed to occurs[4].

### 2.1.9. Absence of software architecture artifact

The development of any software architecture can be complex if standardized way of generating software architecture artifact is missing. To some extent, software architecture based on analysis model is made. [5]

## 2.2. Challenges of Model Driven Architecture

In today's time, our software industry has become very challenging. Due to this our software are very complex. Developers always try to maintain the quality of the software by meeting the customer's expectations. A study conducted by Standish group has shown that only 29% of the projects were successful in 2004. Moreover, in typical SDLC used in companies or organizations; it can be easily observed that routines are repeated in terms of same steps, same tools, and tests. To improve these routines, the model-based approaches are vastly used. [6]. Figure 3 illustrate the challenges of MDA these are explained below:

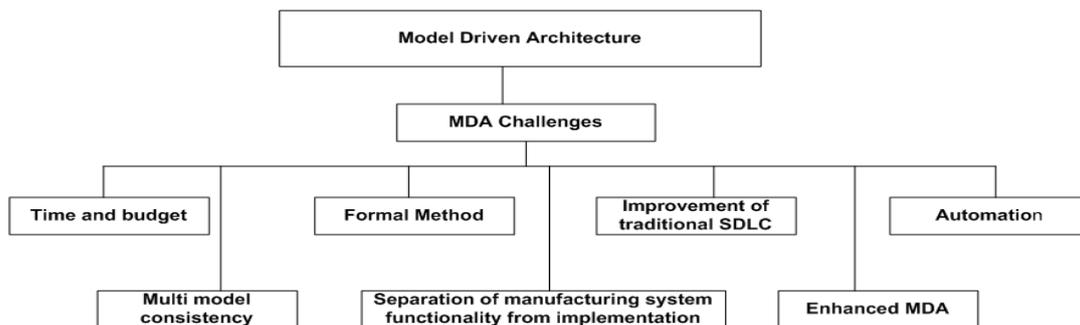


Fig. 3. MDA challenges.

### 2.2.1. Time and budget

MDA development is designed to reduce the cost and delays at each stage of SDLC. It is fundamental if we want to make our software profitable. As many tools and transformation took place in MDA so it is considered a challenge.

### 2.2.2. Formal method

In software development, MDA explains the solution to the problem by providing only formal methods. No other method is used.

### 2.2.3. Improvement of traditional SDLC

MDA is a further step to improve the traditional SDLC methods in each step starting from requirement analysis, specification, design, coding, and maintenance. MDA is a set of frameworks developed by OMG. Some of the challenges of MDA are to identify application of transformations to abstract descriptions, further it involves refining that description and make an alteration among dissimilar representations. [6]

### 2.2.4. Automation

To develop an architectural design, it is an important step in MDA which depends on architect's experience and resource constraint. Due to this purpose, automated methods are unachievable yet. [7]

### 2.2.5. Multi model consistency

Another important challenge in MDA is to understand the relationship between these models that they should maintain a multi-model consistency as changes in one model affect other models in the multi-model.

[8]

#### 2.2.6. Separation of manufacturing system functionality from implementation

It is important to manufacture the system functionality from the specification from which it is implemented on any particular technology platform. It is considered a challenge because it makes the reconfiguration of the system by using the manufacturing progress through the model transformation and transmission of information. [9]

#### 2.2.7. Enhanced MDA

The existing MDA does not handle the enhanced MDA. Enhanced MDA is basically an architecture which exists in order to increase its efficiency and applicability. Also this enhancement phase cope with changes which are required time to time without getting interrupt in present architecture. In this case the overall efficiency is not compromised. [10]

### 3. Literature Review

In this section we are providing the literature review to highlight the work that had been done on MDA and to identify where new contribution could be achieved For real time systems JunliGao Di Li et al [3] presented an OOMDRDP (object oriented model driven rapid development process). This approach comprises four phase's analysis, design, implementation and testing here all phase has micro recursive process based on model driven architecture. UML is used to erect the model. Object oriented model on the computer numerical controller is applied. From program codes to UML models they have endorsed the non-concrete layer of SD (software development), which perform code associability and testing the functions of computer numerical controller on the UML model. And lastly they have validated the CNC schedule ability against the real time system.

Nourchene Elleuch *et al* [5] proposed a technique in which through analysis of model architecture of any system can be generated. To achieve this they have added one more layer in in MDA which is software architecture layer. Analysis model is treated as AIM (architecture independent model) while they treated the software architecture layer ASM (Architecture specific model). They have concluded that through architecture specific model many benefits can be avail like design pattern different styles of architectures and important decision about design can also be taken. In 2009 Oksana Nikiforova et al [6] proposed a research in which they have analyzed the different CASE tools that are deployed under the MDA process. They have discussed transformations among MDA and MDD (model driven development) in perspective of instinctive competencies for SDLC. The vital impact of this research is CM (component model) for Model Driven Architecture and Model Driven Development which stipulates activities that are supported by the MDA tool chain to mechanize the Model Driven Development process. They have also deliberated how development of the model driven software can be defined by the diverse tool chain. They have also generated the class diagram as well as source code form the application of two models which are hemisphere in nature.

Daniel Perovich *et al* [7] apply the model driven engineering techniques to achieve the architecture design that enables the methodical and aided erection of the Software Architecture applicable of enterprise applications. The describe architecture is cured as a mega model. To complete SA (software architecture) from scratch the architectural basis is unequivocally enumerated as the set of alterations. Their proposed approach consider mega model then comprehend it transform it to system that is well structure and independent and further more express it in a specific language. Claudia Szabo [8] conducted a research in which they have presented the architecture which is based on multi modeling. Relationship of model at different level is internment through this architecture. In order to analyze the change impact on diverse environment taxonomy has been defined. This taxonomy is based on RCMM (relationship correspondence

meta-model). Through this, stakeholders are notified against any change. Their proposed architecture is validated through the case study and they have also emphasized on the challenges that are faced by their proposed approach. Hailong Huang et al [9] conducted a research in which they have presented a framework for dynamic reconfigurable process especially for manufacture enterprises. In order to resolve complications of movability, evidence in corporation and interoperability Dynamic reconfigurable manufacturing process model can be construct with Model Driven Architecture methodology. Specific technology platform implementation can be separated from its manufacturing system functionality specification through the dynamic reconfigurable process, and creates the system recon figurate the industrial advancement over the alteration of models and communication of information. To support the dynamic reconfiguration manufacturing process PIM was erected with UML which is based on the MDA methodology. Many PSMs was transformed though PIM. Hence they prove that the system is applied and operative using the MDA. In order to enhance the efficiency and applicability of the already existing MDA architecture the research is conducted by AtikaQazi et al [10]. Yashwant Singh and Manu Sood[11] elaborated different issues of software development and they are addressed using MDA technologies using separation of concerns through the model, mapping and transformation. These issues include changing technology, multiple platform environment, interoperability, serviceability and changing requirements.. Many challenges issues are found in MDA. Results

## **4. Results**

### **4.1. Proposed Solution for Issues**

More interest is gained by model driven architecture now days. We can solve many problems creatively using MDA. MDA is not work in detail and it is still at high level of vision. Models are used to build the large-scale solutions. A lot of issues are faced by MDA. Some of those factors are highlighted in the aforementioned table which were found in research papers [1]-[5]. In Table 1, we are intended to endow with a beneficial analysis that will emphasize on the fundamentals Issues associated with the MDA testing and their possible solution. First issue is MDA model types PIM and PSM which is not integrated in the MDA, as MDA supports different platforms, ides and methodology it must address this issue of model type. Second issue is concrete process inclusion. Most of methodologies claimed to be MDA based but they only support only generic process needs not MDA concrete process. MDA development process is different from the traditional software development process this needs to incorporate in the MDA architecture. Third issue is coverage of life cycle. MDA development life cycle is different from the traditional development life cycle. Traditional software development life cycle involve different phases like requirement election, software specification, design, coding, testing and maintenance. Which of these phases should be converged in the MDA is crucial issue. Fourth issue is application scope. Development process should be defined effectively and efficiently by the people involved in the development process. If the development process is not precise and clear then a lot of issues can occur and different interpretation can be made by the people involved in the development process. MDA define development process but not in effective way. Fifth issue is Application scope. Application scope defines the domain of the particular application. Different applications have different scope. Some MDA methodologies are not general purpose and they are not applicable to all domain. Sixth issue is security. Software security assurance is a procedure that assistances design and implements software that defends the data and resources enclosed in and controlled by that software. Seventh issue is real time system support and development. As MDA support development of real time systems but it is partial. All functions of real time systems are not supported by the MDA. Eighth issue is non-functional requirements. It is difficult to represent NFR. There is variety of issues in satisfying the non-functional requirements in MDA such as modeling of non-functional requirements and their

relationships, level of satisfaction of NFRs at run time. To deal with such issues, system needs to be monitored during runtime and if there is any violation, system requires reconfiguration by introducing new functional properties. Ninth issue is absence of software architecture artifacts. As MDA is architecture. We might face difficult is development of software architecture is consistent way of generating software architecture artifact is missing.

In Fig. 4, we have proposed a plug-in throw which all of the above mentioned issues can be resolved and incorporated in the existing MDA architecture. Our purposed plug-in identify the root cause behind any issue and reduces chances of issues occurrences. Issues plugin check and validate the each step of software development life cycle to assure that right each of the above mention issue is correct before the implementation of MDA application. Working of our purposed plugin is described throw the following diagram. In this diagram project initialization phases basically encompass the MDA support Phases. These phases consists of platform and model Identification and specification, process identification & specification, modeling languages identification & specification, transformation identification & specification , tool selection, application scope identification & specification, security level & real time system identification & support and NFR requirements specification. Under MDA architecture when we start project initialization MDA support phases sustenance our proposed MDA issues plug-in by mapping our MDA issues with the MDA support phases and enabling those issues to be reduced efficiently and effectively. Our proposed MDA plug-in not only resolve those issues throughout the software development and also increase the efficiency of the MDA existing architecture.

Table 1. MDA Issues

Reference No	MDA model types	Process inclusion	Coverage of lifecycle	Process Precision	Application scope	Security	Real time system	Non-functional requirements	s/w architecture artifact
1	✓	×	×	×	×	×	×	×	×
2	×	×	×	×	×	✓	✓	×	×
3	×	×	×	×	×	×	✓	×	×
4	×	×	×	×	×	×	×	✓	×
5	×	✓	✓	✓	✓	×	×	×	×

Our proposed plug-in is not feasibly for the MDA products in term of its applicability on different MDA projects also this plug-in is cost effective in prevention of maintenance cost after the deployment of specific MDA product which could be increase due to presences of those issues. This plug-in is best suited for large scales projects as it increase the overall performance of the MDA architectures by not only resolving those issues also enhancing the incentives of MDA architecture.

**4.2. Proposed Solution for Challenges**

In Table 2, we have analyzed the papers [6]-[10] and discovered the challenges facing MDA technology, although lot of work is done on these to cater but these challenges are still problematic. If we consider time and budget, it is the most important factor in any framework. It indirectly affects the performance. So, it should be less time and cost consuming but MDA composed of many phases which take time and cost. Formal method should be adopted in order to remove ambiguity in any phase of the products being developed but in some cases informal methods should be supported as it can be requirement of the system. Automation is an important parameter considered here and it is not achieved yet by using MDA. As it is

clearly defined above the structure of MDA that it consist of multiple models which need to be consistent. If consistency is not found then interface and interaction problems can be observed. Separation of system functionality from the implementation is a challenge to achieve due to the reconfiguration involved by the transformation. Enhanced MDA's technology should be adopted it would be helpful in achieving the efficiency and applicability of MDA process. However, it is necessarily to note that MDA itself is a relatively new approach. Therefore, there should be clearance in appropriate techniques, tools, and methods, which of them are working, which are not and why.

In Fig. 5, solution to all these problems can be provided in the form of plug-in provided with the software architecture tool. Many software architecture tools exist which include MODA-TEL, MIDAS, MASTER, C3, ODAC and many more. These tools have different features such as MDA support, process inclusion, coverage of generic life cycle, process precision and application scope.

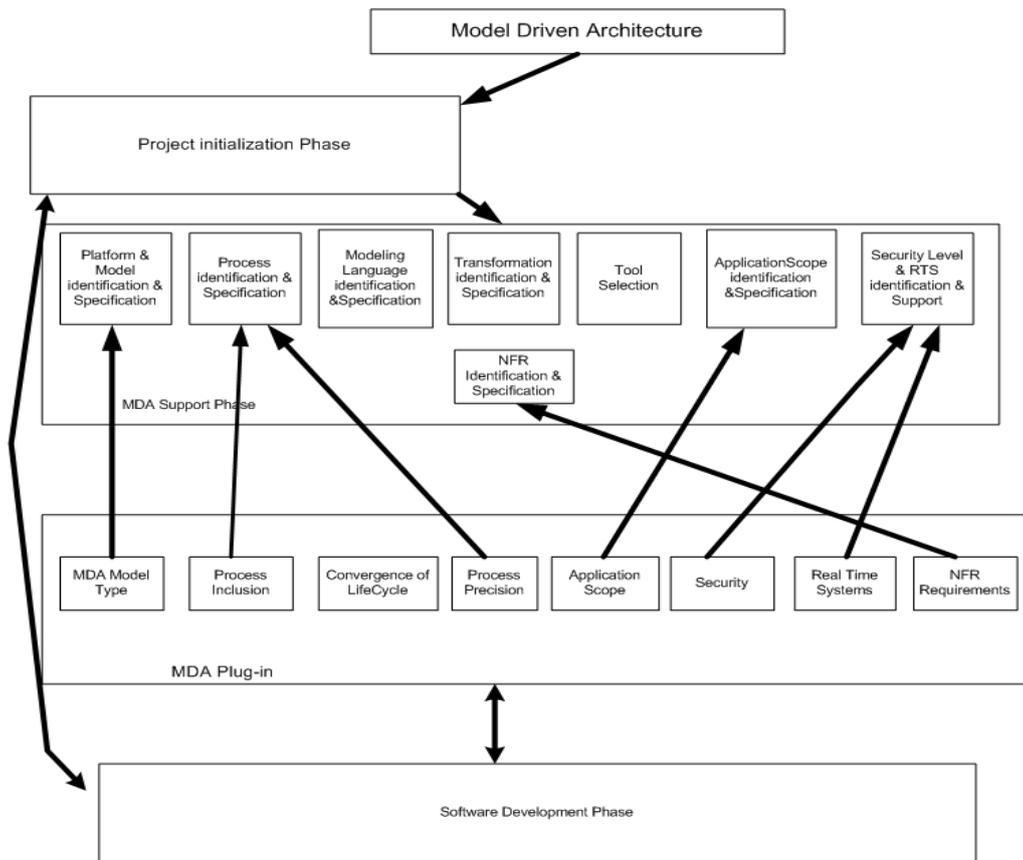


Fig. 4. MDA issues plug-in architecture.

Table 2. MDA Challenges

Reference No	Time and budget	Formal Methods	Improvement in SDLC	Automation	Multi-model consistency	Separation of system functionality from implementation	Enhanced MDA
6	✓	✓	✓	×	×	×	×
7	×	×	×	✓	×	×	×
8	×	×	×	×	✓	×	×
9	×	×	×	×	×	✓	×
10	×	×	×	×	×	×	✓

Different plug-in are widely used for many software and serve for different purpose. Here, in our case some plug-in should be provided as well.

In this diagram, when MDA project is initialized developer seeks for functions provided in the MDA support phase. And these functions are basically challenges of the MDA and not provided yet. In the next step, software development phase needs to install these plug-in for the required functionality. Mapping through lines depicts that which functionality should be supported by which plug-in. These plug-in would be very useful in development area due to number of reasons .These would be easy to find as they will be provided with the software architecture. They are install and feasible to use. They fulfill the functionality required to the developer for the user hence in other words can increase performance. In order to satisfy the customers with the overall functionality prototype can be built using Eclipse and ATL. Some other solution can also be proposed in the form of utility and another framework but MDA is itself a framework so it would not be effective to build another framework. Finally, we can say that the plug-in approach is better than other approaches.

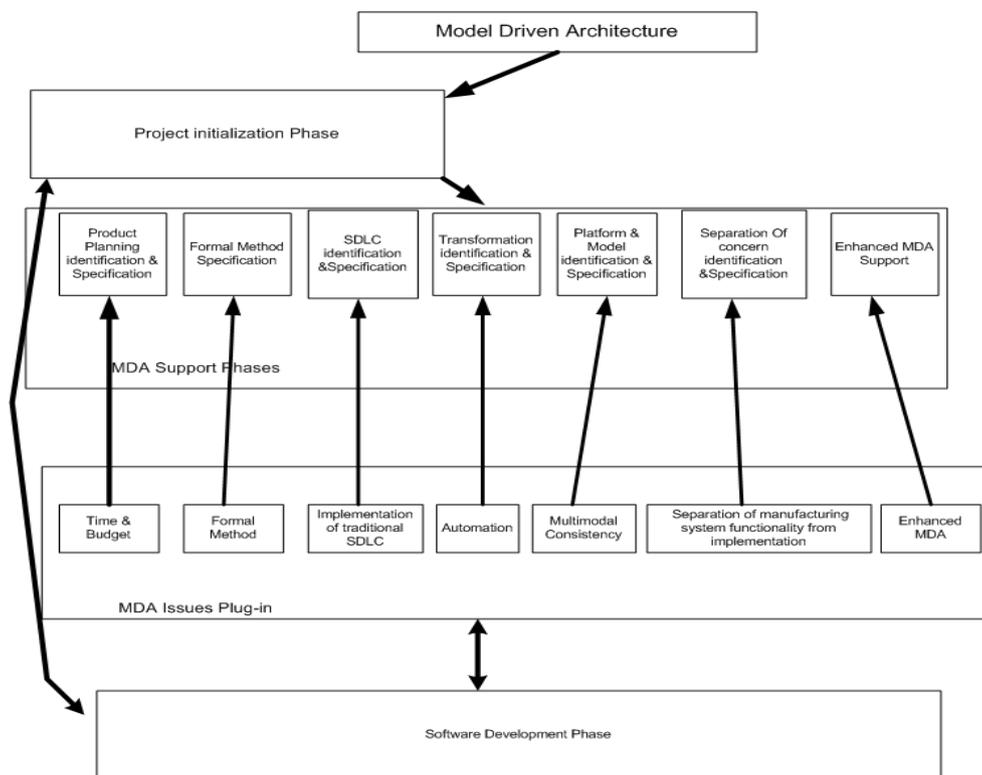


Fig. 5. Solution to challenges.

## 5. Conclusion and Future Work

More interest is gained by model driven architecture now days. We can solve many problems creatively using MDA. MDA is not work in detail and it is still at high level of vision. Models are used to build the large-scale solutions. In this paper we have identified some of the most vital issues and challenges faced MDA existing architecture. We have proposed two novel plug-ins for MDA architecture. These plug-ins are capable of handling all those issues and challenges in effective and efficient way. Thus are proposed solution can is best suited for large scales projects as it increase the overall performance of the MDA architectures by not only resolving those issues also enhancing the incentives of MDA architecture.

Another enhancement in MDA is to develop a computer-aided software architecture design environment that is helpful in dealing with architecture representation. Usability framework to other MDA tools can be

introduced. It will evaluate and extend the usability model which can improve the application development.

## References

- [1] Fatemeh, C., Maryam, Y., & Raman, R. (2007). Methodology support for the model driven architecture. *Proceedings of the 14th Asia-Pacific Software Engineering Conference.*
- [2] Nina, M., Kurt, S., Holger, G., & Wolfgang, R. (2009). Model-driven code generation for secure smart card applications. *Proceedings of the Australian Software Engineering Conference Gold Coast.*
- [3] Gao, J. L., Li, D., & Zheng, S. X. (2006). developing real-time system based on model driven architecture. *Proceedings of the International Conference on Mechatronics and Automation.*
- [4] Kiran, K., & Arvind, K. (2013). Integration of non-functional requirements in model driven architecture. *Proceedings of the Fifth International Conference on Advances in Recent Technologies in Communication and Computing.*
- [5] Nourchene, E., Adel, K., & Samir, B. A. (2007). Software architecture in model driven architecture. *Proceedings of the International Symposium on Computational Intelligence and Intelligent Informatics.*
- [6] Oksana, N., Antons, C., & Natalja, P. (2009). Discussing the difference between model driven architecture and model driven development in context of supporting tools. *Proceedings of the ICSEA '09. Fourth International Conference on Software engineering Advance.*
- [7] Daniel, P., Maria, C. B., & Cristian, R. (2009). Model-driven approach to software architecture design. *Proceedings of the ICSE Workshop on Sharing and Reusing Architectural Knowledge.*
- [8] Claudia, S., & Yufei, C. (2013). A model-driven approach for ensuring change traceability and multi-model consistency. *Proceedings of the Australian Software Engineering Conference.*
- [9] Hailong, H., FeiLiua, B., & Wang, Q. F. (2009). Research on dynamic reconfigurable manufacturing process model based on model driven architecture. *Proceedings of the WRI World Congress on Computer Science and Information Engineering.*
- [10] Atika, Q., Azra, S., & Haseena, B. (2013). Model driven architecture with encapsulated quality check and enhancement feature. *Proceedings of the Third International Conference on Innovative Computing Technology.*
- [11] Yashwant, S., & Manu, S. (2009). Model driven architecture: A perspective. *Proceedings of the IEEE International on Conference Advance Computing.* March.



**Amna Noureen** is a research scholar at College of Electrical and Mechanical Engineering, NUST, Rawalpindi, Pakistan



**Anam Amjad** is a research scholar at College of Electrical and Mechanical Engineering, NUST, Rawalpindi, Pakistan



**Farooque Azam** is a research supervisor at College of Electrical and Mechanical Engineering, NUST, Rawalpindi, Pakistan