Integration of Information systems using Model Driven Architecture (MDA)

Beenazir Ganji
Islamic Azad University
Najafabad Branch, Isfahan, Iran

Mohammad Davarpanah Jazi
Islamic azad university
Najafabad Branch, Isfahan, Iran

Mehran Sharafi
Islamic azad university
Najafabad Branch, Isfahan, Iran

ABSTRACT
No doubt there are difficult problems before IT managers and software development pioneers. These problems are the main challenge for the dutifulness of today modern organizations. Developing Information system requires accurate and tiring work. It should be done by very experienced programmers and needs spending a lot of many on the other hand, today requirements direct us toward extensive systems and Integration of these systems is one of the concerns of researchers. Although it is not much time since MDA introduced to software industry, its standards are not yet completed and introduced
However, we try to use it to develop and integrate in formation systems. The goal of this article is to focus on applying MDA in integration of systems. In fact we want to show how MDA facilities can cover aspects of integration, and how we should use these to reach an appropriate Information system

Key words : Information systems, Model Driven Architecture, Integration, modeling

I. INTRODUCTION
The end of twentieth century changed under influence of three important phenomena:

- Customer, competition, transformation . customer: is not sat is field with every goods and every quality.
- -Competition: due to population increase and extent of globalization, the competition increased to its maximum.
- -Transformation: no customer bugs stable and unchanged services. On the other hand every producer will lose competition it do not have any variety in its goods.

Moreover, in today world full of competition in which organizations and employers and employees face much difficulties from competitions, customers, share holders, an organization or a producer requires following vital components to win the competition:

1- High efficiency and correct planning
2- Controlling products last price
3- Correct management of matters and goods existence
4- Controlling organization commitments and cach

A. kinds of software systems regarding application

- small mono – purpose systems.

These systems are supplied as ready packages which include a brief system performance.

- Island systems

A collection of financial and official systems which operate separately and information transfers between systems by user.

- Information systems (MIS)

The other name of MIS is Integrative Information systems. They are presented as a software pack which has inflexible and integrative structure and cannot cover information requirements of big organizations that have specific and complicated processes.

- Enterprise Resource Planning (ERP)

One of the best definitions for ERP is "An applied software package which is a collection of Integrated Modules ready to start, pre-planned and pre-engineered that covers all Enterprise Business Process."

B. Introducing some Expressions

In this part some of concepts will be defined.

1) Enterprise

An Enterprise includes:

- people, Information and technologies
- it does the job responsibly.
- it has a defined organization structure which is distributed different places.
- It responds to internal and external occurrences
- It has a strategy for its activities
- -It provides its customers and audiences which specifiedproductions or services

2) Architecture

Architecture of a system is a characteristic of parts and relationships between them which includes regulations related to interaction of these parts which each other through applying relationships. One of kinds of Architecture is Model Driven Architecture.

3) Modeling

One of the main fields in computer is software production for computer. The software it self is based on a model for it is a solution of a real problem in computer discrete world.

What is a Model?

We can best define a model as "regarding essential aspects of a problem or phenomenon "Either characteristics regarded or not depends on its modeling angel and detailing standard.

Why we use model?

Model is a picture of system or a part of it. Rely on a model can make our view simple toward system and its details. These models are summaries of projects which do not include information about reaching performance ways.
Today UML models are more used.

- **INFORMATION SYSTEMS INTEGRATION**

Information systems Integration is done for collecting, processing, controlling and saving in formation and making sure of Information progress correctness, facilitating operational out puts and also supporting management information in making decisions with integrated manual and computer components. Integration is a goal, but the ways reaching it is different. There are two main techniques reaching Integration which of course using either of them depends on the level of Enterprise Information systems

maturity. Common technique would be abandoning Island systems and developing an Integrated Information system which is usually known in the literature of IT as Enterprise Resource planning (ERP) that is defined in the previous part. In cases that Integration of the first kind is impossible, it is common to apply the second kind, Enterprise Application Integration (EAI).In this technique instead of abandoning current Enterprise Information systems. We struggle to reach Integration via connecting systems. The goal of this method is to hold data, software and organizational processes common efficiently and integrative of an Enterprise Basis, and relationship between Information, software and diverse Enterprise Applications associates a new virtual environment . Researchers have paid attention to this technique from 2004.

A. The goals of applying Integrative Information systems managers of Enterprise and Institutes follow extent goals through applying Integrative Information systems(most common Enterprise Resource planning or other Integrative Information systems), but the most important goals about settlement and applying IIS are:

a) To perform main and frequent activities in an intelligent way and essentially in operational outputs;

b) To exert internal controllings on main activities in an intelligent way;

c) Fast availability to classified information for making decision;

d) To save time in information processing;

e) To minimize humane mistakes and to prevent redoing and wastes;

f) To prevent exerting personal tastes in performing main and frequent Enterprise activities;

g) To facilitate collecting and saving information;

h) To prevent spending repeated times in producing information about same subjects;

i) To increase effieacy and efficiency in Enterprise;

j) To help users with little skills;

B. shortcomings and Difficulties of Integrative Information system some of shortcomings and difficulties related to Integrative Information system are:

1- In spite of essential sensitivity and differences between software projects management and general projects management, computer enterprise and institutes do not take sufficient care in choosing plan projects manager and settlement of Integrative Information system, and simply neglect this important matter which may lead to project failure.
2- some of these institutes with planning apart of financial non correlated systems of organizations and institutes (book keeping, financial , properties, store) or conformity of some prepared computer software with activities of organizations(enterprise) and institutes (institutions) imaging that they've done restoring order and they move in the frame of Integrative Information systems. This incorrect reception and perception of Integrated Information system causes internal controlling not to settle with organization activity basically and appropriately and on the other hand not to meet the information needs of users both in side and out side of organizations.
3- Yet there is an incorrect perception. System planners think they can mechanize current techniques and Traditional (manual) Information systems of organizations and institutes with out any fundamental changes, training and acculturation.
4- Not training system planners and analysts appropriately in "aquiring human relationship skills"
5- In most cases which Integrative Information system plan and settled on Enterprise Information needs and on customers order, people do not pay much attention to system relationships and do not study about Information systems facilities sufficiently and skillfully.
6- Attention and performing the analysis and planning of activates which is possible to mechanize them, and neglecting the analysis and planning of activities which should be done manually but integrative and concordant with other activities.
7- In cases that Integrative Information system is done based on customers needs and order, planner's concern is limited to inform the customer about the process in each planning stage specially in analysis stage and in determining systems constraints.
8- carelessness of system planner and suppliers to the requirements of system startup time and the ways of collecting information and in some cases not cooperating about this.

III. PHILOSOPHY OF MODEL DRIVEN ARCHITECTURE (MDA)

Object management Group (OMG)formed in 1989. This group has been established from several organizations' coalition and its goal is to create standards and to encourage applying object oriented Analysis Design. During 90s, OMG created standards which is totally called object management Architecture (OMA).Central part of OMA is an in ware standard which is called "common object Request
Broker Architecture (CORBA)". To solve the problems in Integration we need a solution more common than CORBA.MDA is the result of efforts to solve this problem. This Architecture is a simply and comprehensive technique which determines standards required to make, integrate and keep software properties. The goal of MDA is to prepare an

Enterprise Architecture modeling that analysis and developers can use it to define a software work or property.

A. model Driven solution

Model Driven Architecture is the natural perfection of UML object oriented Analysis &Design, coders and computing of 3rd millennium. MDA reminds us of the time when UML was software planning and making standard. MDA uses models as a powerful lever to develop software. MDA is not only a development process, it is not only a feature, it is not only download, MDA is not only a reference download, too. MDA defines a frame to process and describe models. MDA gadgets change business models with less technique decisions into quite applicable, settling and performable programs. MDA defines a method Information systems features in which system duties feature has been separated from downloading those duties on a technology special platform. To do this, MDA defines an architecture for models in which a collection of approaches for organizing stated features prepared via diverse models. MDA is to develop system in which models are of great importance. This is a model-oriented model. Because models have the responsibility of directing comprehension flow, planning, making, settlement, exploiting, keeping and reforming. This Architecture techniqually is a frame to process and describe models. Briefly, MDA is an Endeavour to settle model at the exit of software production and not just using models to change code into quite applied, settled and performable programs. This change is done automatically that will be studied at following .

B. main Idea of MDA

Concentrating on modeling renewal and change into software performing (UML model _ java code), software planning method, separating planning from architecture and performance, focusing on needs through modeling instead of performance, simplifying needs change.

C. some of MDA advantages

separating model (needs analysis), portability, connecting platforms, changeability of requests, independence of platforms, omitting manual coding of a model behaviour, planning better and more precise by focusing on models, a revolution in increasing abstraction level, separating work regulations from down load environment.

D. Basic concepts

In this part we introduce basic and first concepts in MDA:

1) model Driven

In MDA method, models have the responsibility to divert comprehension flow, plan, make, settle, exploit, keep and reform the system.

2) Viewpoint

One view point employs a selected collection of architecture concepts and organizing rules to concentrate on special concepts in a system. MDA introduces 3 view points of a system.

- View point independent from computing: this view point emphasizes on system enviornment and its requirement. In this viewpoint structures details and system processes are either hidden or not determined.
- View point independent from platform: this viewpoint emphasize on the operation of a system and required components for downloading on a platform is hidden into In fact, this viewpoint shows a part of system quite feature which stays unchanged from a platform to other.
- View point special for platform: this viewpoint shows the previous viewpoint accompanied by downloading components on a special platform.
-

3) Model Transformation (MT)

MT is said to the process of changing a model of one system into another model from the same system.

E. The cycle of MDA

1) Computation Independent model (CIM) One CIM is a view of system based on computation independent viewpoint. One CIM is a model of system which shows how a system works in environment. So, it helps in understanding exactly what a system does.

2) platform Independent model (PIM)
One PIM is a view of system based on platform Independent viewpoint. This model describes a system, but does not show components related to using platform. One PIM includes features of organization views, information and computiations.

3) Platform specific model (PSM)

One PSM is a view of system based on platform specific view point. One PSM combinates PIM in side features with components relates to how to use system on a specific plat form. Today this model is usually as a software & hardware guiding booklet (manual) or is even sometimes only in Architect's thought.

4) Mapping

In the next step a mapping should be prepared to transform a PIM to a PSM. In this process, platform model determines mapping identity.

Kinds of mapping:

1. Model Type mapping:

"model Type" mapping is one mapping made of each model in which mapping is done through specified kinds in PIM language (like class, dependence relation, or other components of a model) to corresponding kinds in PSM.

1. Model Instance mappings

Another way for mapping models is to identify components of PIM model which should transform in a specific way to reach PSM model. "sample model" mappings define "marks". One mark shows a concept in PSM, and is exerted on a PIM component to show how that component should transform. These marks that are a platform specific, are not only a part of PIM, but also added to PIM to enable transforming model to PSM. Architect takes PIM and markes it so that it would be usable for a specific platform. This marked PIM is used later in PSM development.

1. Synthetic mappings of kind and sample

Most mappings are a combination of 2 states mappings. A mapping of model kind has the capability of stating transformation as rules of a kind in PIM to one or more kinds in PSM. Where as, each transformation based on a model sample either clearly states what mark is appropriate for what kinds in PIM or some constraints should be specified implicitly for users of marks.

5) Marking a model

In model sample mapping, the architect marks components of PIM to specify how to use a mapping in transformation of PIM to PSM. In simplest state, PIM components are marked only once and show that there is a definite mapping to transform that component to one or more

PSM components. In a more common state, several PIM components are marked to show their role in same mappings. Then these mappings are used to transform those component to a collection of PSM components which could be different apparently. One PIM component may be marked several times (each with a different mapping) this shows that component has roles in diverse mappings. When a component is marked like this transformation is done according to each mapping.

6) Transformation

The next step is to take marked PIM and transform it to a PSM This could be done either manually or automatically. W indicated that transformation is the process of changing a model to the other model of the same system. Input of this process is a marked PIM and a mapping. Output is PSM and transformation record. If we use model kind mapping, the process will take a PIM, perform mapping on it and then produce a PSM. When we use Model Instance mappings, the process receives transformation of a marked PIM, does the mapping according to marks and at last produces a PSM. Some of facilities may transform a PIM directly to a settled code (without first producing a PSM). However, its better to produce such PSM devices to comprehend and decode easily.

7) Transformation Record (TR)

The results of transformation process of a PIM are :"a PSM" and "TR". TR includes a mapping of PIM components corresponding to components in PSM that shows which parts of mapping are used for each part of a transformation process. This record is prepared to people who mark on PIM or PSM. MDA modeling facilities which keep TR, may coordinate PIM and PSM in changes.

8) Preparing PSMs

The PSM which is produced through a transformation process is a model of that very system which is specified by PIM. This model shows how system takes use of selected platform. A PSM could have less or more components based on its duty. PSM could bean "implementation" if it prepares all information required for making system and operating it. On the other hand PSM can operate as a PIM that only is used for more refining and developing a PSM which could be implemented directly. If PSM would be an implementation, it shows a wide spectrum of information. These information may include: program code, kinds of implementation related to CORBA of connectors features, settlement definers , configuration features , ….

IV. STANDARDS AND MODEL DRIVEN ARCHITCTURE

META MODELS

object management Group has employed some of different standards and meta models to form model driven Architecture with their cooperation. Here we introduce some of the most important facilities.

A. Meta- object facility (MOF) :

MOF is a frame to define, manipulate, integrate meta data & data. This is done by an independent technique of platform. MOF feature lets models to get out of an applied program and enter the other; to transfer through different webs, to be saved and restored in a warehouse and transform to diverse shapes. Standards based on MOF are employed for integration of programs and data. All models and meta models used in MDA are defined based on MOF which are said to be the MOF compliant. So, their integration in all of modeling process is possible.

B. Unified modeling Language (UML) :

UML is a Graphical modeling language which is used to picture, document and limit out puts of object oriented Analysis & Design.

C. XML meta data Interchange (XML)

XML defines an Interchange frame based on XML which is used for UML models and meta models or any MOF – compliant. Also, this standard prepares a mapping between MOF and XML.

D. Object constaint language (OCL) :

OCL is a standard aid for UML. This standard lets you to specify constraints and logic in using models. OCL is a formal language which defines expressions in UML models.

E. common warehouse meta model (CWM)

CWM is a comprehensive meta model that enables data mining in data bases of an enterprise. The position of CWM in data modeling is the same as UML position in programs modeling .

F. Web services meta model (WSM)

This meta model has been developed based on MOF to facilitate developing web service models. Web service is a modular Applied program and reusable which is used under web.

G. UML themes

By themes we can make UML language ordered for a specific area of computations (ex.Distributed computations) or a specific platform (ex. EJB or CORBA).In MDA, PIM& PSM models are defined with themes.

V. MDA SOLUTION

As stated before, there are different challenges facing development of MDA based information systems. At first
we state integration challenges and then we show how MDA can solve these problems.

Challenge 1: software industry has special characteristics that separate it from other industries. Every year (or even faster) new technologies are innovated and developed quickly (ex. Java, Linux, UML, SOAP, HTML, XML, ASP, JSP, Net, J2EE, web services , ….). most of companies should coordinate with new technologies and move with them. It is worst in integration. Because even if you just focus on a specific technology, you need coordination with them to interact with other software and to integrate them. Large scales in recent systems makes inevitable using a wide spectrum of technologies. How can we unify these technologies?

Solution 1: MDA can be useful in integration of information systems through separating different models in different individual levels. When a technology changer, PIM& CIM stay unchanged and only PSM models should change. Indeed just PSM models change is more simple and cheaper than all software change.

Challenge 2: There are new technologies that are not yet completely developed, but in future they develop. How Information systems integrate with these technologies?

Solution 2: Here we feel the necessity of a common language to understand development models of a software for integrating it. MDA has solved this problem by presenting a MOF standard. If new technologies develop based on standard, the MOF – compliant software can easily interact with them. So, it is easier to unify them than the time there is no common meta model.

Challenge 3: production and updating documents is useful to improve keeping and integration of software. Are documents produces in information systems (IS), regarding IS large scale? How we can produce and update these documents?

Solution 3: Documentation is always one of software Development process Weaknesses and is always seen as a lateral

work. Some of developers think that their main work is to produce code writing documents in the process takes time and leads to decreasing the process. Except project manager who forces writing documents, there is no motivation to do so. For this reason qualitieal documents are not produced and problems are made in keeping and integrating software. This is worse in IS. MDA can improve

the condition. In MDA production of models and documents is the main duty and producing code is done automatically. MDA facilities are able to keep and integrate different models by their documents which leads to having sufficient information at the time of keeping, developing and integrating.

Challenge 4: Integration requires a big change in cooperation policies. In IS it is more necessary due to large scale. How we can facilitate cooperation policies?

Solution 4: MDA emphasizes on cooperation and coordination between models based on a standard and a unique method. MOF can prepare a common language for cooperation as the core of interaction and cooperation. Also, facilities and automatic techniques of model mapping can facilitate most of required cooperations. Challenge 5: Integration efforts because of wide area of operation (specially in extend systems) influence on business when the most important duties of a business formed as an integrated solution it is vital for this business solution to do right. How to make sure of their performance before implementation and integration?

Solution 5: In MDA, models are facilities of directing and guiding software development process. It is possible that after analyzing and planning models we perform them to make sure of their performance. Techniques related to Executable UML which are regarded in MDA let people perform and evaluate models directly. In this way it is possible to test, analyze and plan effectively and to make sure of their correctness before implementation and integration.

Challenge 6: Despite great need for integrated solutions, just a few standards developed in this area. These standards are not coordinated too, and lead to disturbance in new development and interpretation of standard. How can we increase cooperation and integration between different "standard compliant" productions?(productions which follow special standards).

Solution 6: MDA emphasizes that all standards should follow MOF.Being four different levels in MOF standard (Mo) to (M3) levels facilitates standards definitions. Since MOF has meta model Definition level we can define new standards which their selves follow above meta standard.

Challenge 7: Today IS are a techno-social system. Their integration requires blending a collection of skills that some of them are not in software engineering and IT area.

Solution 7: MDA do not present any solution for this problem, but potentially can solve this. If we can develop standards and models for other non-IT expertations based on MOF, we will use them in integration of software system.

VI. CONCLUSION

In this paper we studied information systems and their integration .Development of recent systems has led to that recent approaches which used in software engineering are not able to meet their needs. To solve these challenges we should take a new and different approach. MDA is one of newest standards which provides software industry with modern capabilities.We tried to present first solutions for challenges of information systems integration. It is necessary to say that MDA certainly could not be complete response. Regarding challenges of information systems development, we are yet at the beginning of developing these systems. In the following of this research we should regard other challenging areas, and try to present solution for them. Also, we can collect MOF – compliant standards for other non-technological areas to integrate systems correctly

VII. REFERENCES

[1] SEI, Ultra-Large-Scale Systems: The Software Challengeof the Future, Software Engineering Institute (SEI), Carnegie Mellon University, June 2006.

[2] Frankel, David S., Model Driven Architecture: Applying MDA to Enterprise Computing, OMG Press, Wiley Publishing, 2003.

[3] Hogg, John, Applying UML 2 to Model-Driven Architecture, IBM Software Group,October 2003.

[4] Pilone, Dan, Pitman, Neil, UML 2.0 in a Nutshell, O'Reilly Media Inc., 2005.

[5]Sacevski, Igor and Veseli, Jadranka. Introduction to Model Driven Architecture (MDA), June 2007

[6] Brown, Alan, "An Introduction to Model Driven Architecture: MDA and today's systems," IBM Rational Edge, January 2004.

[7] Mellor, Stephen J., Scott, Kendall, Uhl, Axel, Weise, Dirk, MDA Distilled: Principles_ of Model-Driven Architecture, Addison Wesley Professional, 2004.

[8] Barry, Douglas K., Web Services and Service-Oriented Architectures: The SavvManager's Guide, Morgan Kaufmann Publishers, 2003.

[9]Ostadzadeh, S.S., Shams, F., Ostadzadeh, A"Employing MDA in Enterprise Architecture,"Proceedings of the 12th International CSI ComputerConference (CSICC'07), pp. 1646 1653, February 2007.

[10] Object Management Group, Common Object Request Broker Architecture(CORBA): Core Specification, OMG Document formal/04-03-12, March 2004.

[11] Object Management Group, MOF 2.0 IDL pecification, Version 2.1, OMG Documentptc/04-07-01, July 2004.

[12] Object Management Group, Object Constraint anguage (OCL) SpecificationVersion 2.0, OMG Document formal/06-05 01, May 2006.

[13] Object Management Group, Common Warehouse Metamodel (CWM) SpecificationOMG Document formal/03 03-02, March 2003

[14] Object Management Group, MOF 2.0/XMI Mapping Specification, Version 2.1, OMGDocument formal/05-09-01, September 2005.

[15]Adel Alti, Tahar Khammaci and ,Adel Smeda, Integrating Software Architecture Concepts into the MDAPlatform with UML Profile, 2007 Science Publications

[16]MDA Guide, OMG document, www.omg.org