Model-driven methodology in design and implementation of E-government

Yashar Eskandari¹
Reza Maleki²
Ali Maleki³

Abstract

In this study, E-government which is considered a fundamental structure in most countries, has been used as a complex and diverse software to explore one of the most advanced methodologies in software management, Model-driven methodology, in design and implementation of such softwares, taking all the requirements, applications and architectural conditions, in to account. Model-driven is in fact the latest step in rising abstraction level in designing an implementation of software which was started many years ago by introduction of functions and procedures in programming languages and then was progressed significantly by the birth of object oriented programming. After explaining the specifications of this methodology and the E-government architecture, applying this methodology will be studied in designing said architecture.

By doing so, specific requirements of E-government architecture will be revealed and defined. Then, by showing fulfillment of these requirements through model-driven methodology, its applicability will be justified.

Keywords – E-government, model-driven architecture, software platforms.

1- Introduction

At first, software design and development methodologies and procedures were identified by features like speed or adaptability, but today most methods have a combination of these features. One of these methods is called model-driven which was developed by OMG [1] organization. This method uses models in different steps of design and implementation, in order to develop very complex systems without any designated platform, by increasing the abstraction levels.

In this study we will try to take a step in direction of an integrated E-government as a complex system and model driven methodology will be introduced and tested as a fitting method of executing this system.

2- Model-driven architecture (MDA)

MDA is one of latest methods in software industry which includes applying abstraction levels in design stage and automated code generation using objects obtained in design stage.

¹ Department of Management, Islamic Azad University, Arak Branch, Arak, Iran, Yashar.Eskandari@Gmail.com
² Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences(IASBS), Zanjan, Iran, rMaleki@iasbs.ac.ir
³ Department of Management, College of Management, Isfahan Science and Research Branch, Islamic Azad University, Isfahan, Iran, Ali.Maleki100@Gmail.com
In this method, high level models shape the foundation of software and then these models are mapped to lower levels of abstraction using automated transformations based on specified rules. This process will continue until the code is generated. In some cases all of the code will be generated as specified, and in other instances some manual modifications are necessary.

Model-driven architecture is a set of standards and procedures used for designing software systems. Transformations are the key component in these standards and procedures [2].

Three main levels and one peripheral group of models are considered in development of a system's models:

- Computation independent model (CIM): system requirements are modeled as a CIM. CIM describes the conditions in which the system will be used.
- Platform independent model (PIM): A PIM describes the system without any specifications of the platform on which the system will run. This model then can be prepared to be used with different platforms.
- Platform Specific model (PSM): PSM is the result of mapping a PIM to a platform specifications based on a platform meta-model.
- Meta-models: A meta-model consists of rules and procedures needed for mapping one type of models into another. Meta-models are mapping guidelines in MDA process. For example, after choosing a specific platform, mapping from PIM to PSM will be carried out according to the meta-model of that platform. Meta-models usually use descriptive languages like UML (Fig. 1). [3]

Using model driven method makes design and Implementation of central systems with the ability of integrating current systems, possible. Utilizing this method, different components of a main system can be prepared for different technologies and platforms [2].

Fig 1) Mapping a model based on transformation rules

3. e-Government

- "e-Government is defined with the use of Internet and world wide web in order to introduce government based information and services to citizens." [4]
- Citizens rarely interact with government related organizations voluntarily. State and official procedure often mean long commutes and time consumption. This means that connecting with
the State entities through internet, can be highly beneficial for citizens. Computer networks will be a natural part of environment for future generations. High rates of internet access among teens and youth puts more emphasis on this fact. The increase in internet penetration rates and easier access will result in more demand for internet based services.

e-Government like any other system has advantages and disadvantages. Major disadvantages are:

- Surveillance: One of the major concerns about e-government is the total surveillance of many personal activities by the government.
- Cost: Successful implementation of e-government will lead to extreme cost reductions in many fields; but the implementation process requires heavy resources and investment.
- Accessibility: The e-government services should be easily accessible by every citizen. Limited access will compromise the whole system’s integrity.
- Security risks: The network based nature of the e-government system brings about the same security issues with all the network based systems involving critical data and interactions. Electronic attacks such as hacking, viruses and malwares can cause disruption of data, disclosure of secret information or propagation of false information. [5]

But there are also advantages:

- Process simplification: Using computer technology can simplify official processes drastically. Access to guidelines and helpful information will become more convenient.
- Public involvement and voluntary engagements: One of the main goals of e-government is to encourage more involvement by the public and more voluntary interactions were necessary.
- Environmental benefits: Many environmental goals can be achieved through implementation of e-government, like minimizing paper use, reducing urban traffic and so on.
- Efficiency: With an e-government in place, citizens can perform most of their official interactions at any time or place with highest response rates, only using a computer and network access.

But in current situation in which different departments and several organizations are equipped with their own IT systems, connection of a central system to these auxiliary ones, each with different technology and platform, would be the main challenge to achieve a centralized and monolithic e-Government.

Redesigning and reimplementation of all these auxiliary systems requires massive resources and it is not financially justifiable. So a flexible system which has the ability to coordinate and integrate this complex group of different systems is needed.

4- Conclusion

To achieve an e-government centralized system, high flexibility and acclimation with different platforms and technologies is needed.

Model-driven architecture provides this feature. In addition to this important feature, model driven method has several other compatibilities that can be summarized as follows:

The basic requirements of e-government system are classified in four main categories:
1. Performance, Expansion capability and Versatility

The system must be designed and implemented such that would be able to eliminate commercial requirements and development and adaption based on commercial needs changes. Softwares designed and implemented by model driven approach include independent parts and are highly portable.

2. Platform Independent and using nonspecific technologies.

This feature shows portability and integration ability between platforms by using standard protocols, programming languages, firmwares, development tools and databases.

3. Data Exchange, Coordination with other softwares

By using appropriate interfaces and firmwares, system and other softwares can be connected without the need of new programming and low-level interfaces.

4. Ability of maximizing target network and security

E-government should be able to associate with any new technology and standard without need of fundamental changes. Model driven approach uses partitioning feature and make different viewpoints which help systems adapt with changes easily.

Figure 2 illustrate mentioned features.

Figure 2) Coordination of e-Government features and model driven approach

Figure 2 illustrates required features of e-Government that is Correspondent with model driven approach. So model driven approach can be an appropriate tool for designing and implementing e-Government because:

1. By Increasing the level of abstraction, design of wide systems will be enabled
2. Possibility of reuse in systems designed by model driven approach facilitates development and transmission of components of e-Government system to different parts.
3. System classification to separate and independent applications facilitates integration, change and development.
4. Different viewpoints in Model-driven approach cover complex requirements of e-Government system
5. Different viewpoints and classification in system, helps applying security strategies and also facilitates optimal security as the primary requirement of e-Government.

References