Chapter 10

Towards a Model-Centric Approach for Developing Enterprise Information Systems

Petraq Papajorgji
Center for Applied Optimization, University of Florida, USA

Panos M. Pardalos
Center for Applied Optimization, University of Florida, USA

ABSTRACT

This chapter aims to present a new modeling paradigm that promises to significantly increase the efficiency of developing enterprise information systems. Currently, the software industry faces considerable challenges as it tries to build larger, more complex, software systems with fewer resources. Although modern programming languages such as C++ and Java have in general improved the software development process, they have failed to significantly increase developer's productivity. Thus, developers are considering other paths to address this issue. One of the potential paths is designing, developing and deploying enterprise information systems using the Model Driven Architecture (MDA). MDA is a model-centric approach that allows for modeling the overall business of an enterprise and capturing requirements to developing, deploying, integrating, and managing different kinds of software components without considering any particular implementation technology. At the center of this approach are models; the software development process is driven by constructing models representing the software under development. Code that expresses the implementation of the model in a certain underlying technology is obtained as a result of model transformation. Thus, the intellectual investment spent in developing the business model of an enterprise is not jeopardized by the continuous changes of the implementation technologies. Currently there are two main approaches trying to implement MDA-based tools. One of the approaches is based on the Object Constraint Language and the other on Action Language. An example of designing, developing and deploying an application using this new modeling paradigm is presented. The MDA approach to software development is considered as the biggest shift since the move from Assembler to the first high level languages.

DOI: 10.4018/978-1-61520-625-4.ch010

INTRODUCTION

Business environment is very dynamic. On the one hand mergers and acquisitions impose changes in the business model and therefore changes in the information system. On the other hand, the introduction of new technologies forces constant change in the business environment. In this case the same business model has to be rewritten using a new implementation technology. In order to survive, businesses must have the ability to thrive in a continuously changing and difficult to predict environment. Therefore, it is important to know what aspects of the business are more likely to change and what are not when dealing with constant change.

An important part of the efforts in developing a new information system are spent in mastering the technological complexity of the solution (a particular implementation technology such as Java or .NET or other technology) rather than focusing on understanding the problem to be solved (Pastor & Juan Carlos Molina, 2007). Experience shows that most of the time it is the implementation technology that changes rather than the business model. Therefore, it is desirable to have a software engineering approach that separates the business knowledge from the implementation technology so they can continue to develop without necessitating a complete rework of existing systems (Papajorgji, Clark & Jallas, 2009; Pastor & Juan Carlos Molina, 2007).

The Model Driven Architecture (MDA) is a framework for software development defined by the Object Management Group (OMG) (Object management group [OMG], 2009; OMG Model driven architecture: How systems will be built). The OMG defines MDA as "*Fully-specified platform independent models can enable intellectual property to move away from technology-specific code, helping to insulate business applications from technology evolution and further enable interoperability*" (OMG Model driven architecture: How systems will be built). As presented by OMG, models should be provided with behavior to be able to be executed and therefore tested. At the center of this approach are models; the software development process is driven by constructing models representing the software under development (Papajorgji, Clark & Jallas; Pastor & Juan Carlos Molina, 2007). Code that expresses the implementation of the model in a certain underlying technology is obtained as a result of model transformation.

The MDA approach is often referred to as a model-centric approach as it focuses on the business logic rather than on implementation technicalities of the system in a particular programming environment (MDA Guide V1.0.1, 2009). Thus, the focus of OMG is to take advantage of new technologies and make the application development process independent of the infrastructures they use. Once the model is constructed, it can be transformed into code in several languages (Pastor & Juan Carlos Molina, 2007). The MDA approach to software development is considered as the biggest shift since the move from Assembler to the first high level languages.

An information system is regarded in MDA from three different perspectives: computation independent, platform independent and a platform specific. The computation independent perspective focuses on expressing the business model (business processes, stakeholders, departments and the relationships amongst them) without any consideration about the information system that will be used during the implementation phase. Efforts to represent the business model independently of the computing platform are referred to as **C**omputational **I**ndependent **M**odel (CIM) (Gasevic, Djuric & Devedzic, 2006; Kleppe, Warmer & Blast, 2003; Object Management Group, 2009). A CIM represents a high level model that describes the relationships amongst principal elements of the business model. As an example, a simple model expressing relationships between the manufacturing sector and the sales department can be considered as a CIM. Usually