

Faculty of Computers &
Information,
Cairo University

A Research Topic Proposed For M.SC. Candidate

Research Topic
“Towards Developing eXecutable Business Processes”

By
Ali Abdel-Aziz Ali

Supervisors
Prof. Dr. Ibrahim Farag Dr. Amr Kamel

Towards Developing eXecutable Business Processes

MASTER THESIS PRPOSAL

1. Introduction

As the complexity of the required software grows, it became evident that modeling data and application is a must. Before modeling it was difficult to maintain or modify an existing system. At the earliest stages, UML modeling was of little benefit except that this model provide high level view for the application. After UML became the industry De-facto standard, supporting tools start providing features like generating simple prototype from the model “generating class template with functions prototype”. However, synchronization problems between the model and generated code. Developers have to manually ensure that changes in the code are reflected in the model and vice versa. Further research work in this field allowed current UML tools to generate prototypes as well as auto synchronizing between the code and the model. They typically have proprietary code generation systems with a fixed meta model. Furthermore, the Object Management Group (OMG) devised the Model Driven Architecture (MDA) concepts [1] to allow the definition of data models, automation of generation as well as easy integration, maintenance, testing and simulation of the software system.

Tools and frameworks supporting MDA, such as ANDROMDA [2], employs some parts of MDA that allow the code generation and maintenance from data model. Unfortunately, these frameworks needs high technical skills to benefit from. To overcome this problem, practitioners have developed models for specific application domain; eXecutable Business Process (XBP) is a prominent goal for such an effort. The ultimate objective of the XBP is to develop tools to support business analysts in the same way CAD/CAM tools are used to support engineers [3]. Our goal is to realize the layout for the eXecutable Business Process Framework (XBPF), identify the requirements, integration required and specify the required steps for making/creating this framework to realize this objective. We propose an architecture consisting of 8 different components to realize this system, (As shown in figure 1-1). The proposed system consists of:

1. a business process modeling tool.
tool that support the modeling of business process.
2. a MDA framework.
allow the definition of data models, automation of generation as well as easy integration, maintenance, testing and simulation of the software system
3. a work flow management engine.
describes business processes in a common dialect that lets business people and developers speak the same language.
4. a reverse engineering module.
extracting information from existing systems to be used in modeling the integration with other systems.
5. an Enterprise Service Bus (ESB).
integrating applications, coordinating resources and manipulating information.
6. an ESB UML meta model.
description that defines the structure of the ESB UML integration models.
7. a transformation engine.
manipulate the business process model.
8. an interface tool that integrates this components.

a wizard tool for integrating the previous components.

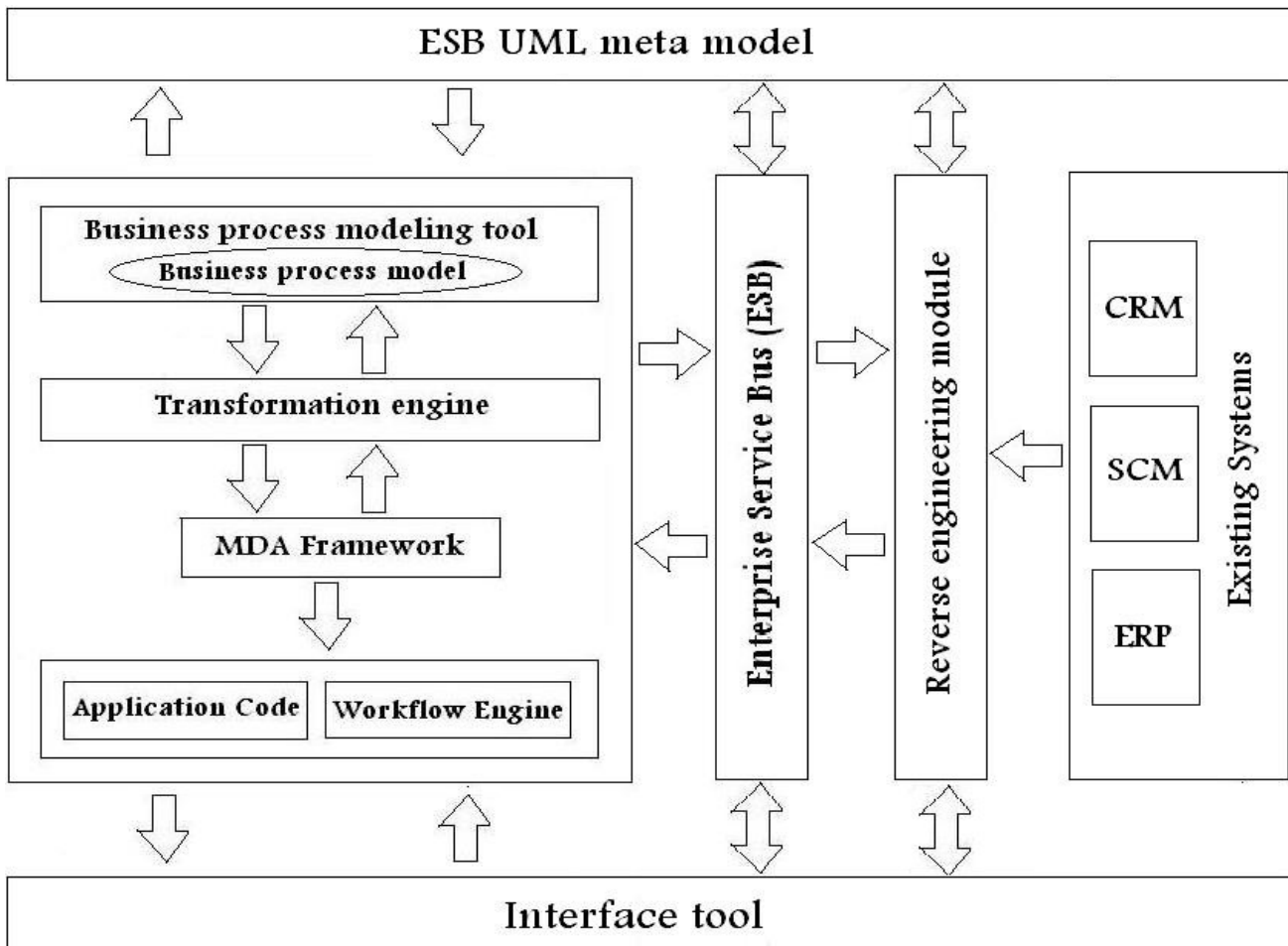


Figure 1-1

2. Background

1. Business process modeling tool:

In 7, November, 2002 the Business Process Management Initiative (BPMI) [4] published the Business Process Modeling Language (BPML) specification document. The Business Process Modeling Language specification provides an abstract model for expressing business processes and supporting entities. *“BPML defines a formal model for expressing abstract and executable processes that address all aspects of enterprise business processes”* [5]; adding to this activities, transactions, data management, concurrency, exception handling and operational semantics. BPML grammar form is XML Schema that enable the persistence and interchange of definitions across heterogeneous systems and modeling tools. Business Process Management Notation (BPMN) 0.9 [6], is a sister specification to BPML 1.0. BPMN aims to be able to generate execution definitions (BPML4WS) [7] that will be used to implement the business processes. BPEL4WS recognizes the need for an independent representation of the interactions between parties.

the following are some names of tools supporting the modeling of business process:

- Magic Draw, ProVision, Enterprise Architecture, and Rational Rose Modeling.

2. MDA framework:

In 2001 OMG adopted MDA framework, the Model Driven Architecture or MDA is an approach to using models in software development. *“The Model-Driven Architecture starts with the well-known and long established idea of separating the specification of the operation of a system from the details of the way that system uses the capabilities of its platform”*. [8]

MDA provides an approach for :

- specifying a system independently of the platform that supports it,
- specifying platforms,
- choosing a particular platform for the system, and
- transforming the system specification into one for a particular platform.

The three primary goals of MDA are portability, interoperability and reusability through architectural separation of concerns.

there exist some good MDA frameworks one of them is ANDROMDA. ANDROMDA will be our MDA framework. AndromDA is an open source MDA framework - it takes any number of models (usually UML models stored in XMI produced from case-tools) combined with any number of AndromDA plug-ins (cartridge and translation-libraries) and produces any number of custom components. You can generate components for any language you want, Java, .Net, HTML, PHP, anything, just write (or customize an existing) plug-ins to support it.

the following are some names of other MDA frameworks:

- ArcStyler, OptimalJ, OpenArchitectureWare, KennedyCarter, Xcoder, Iqgen, and Home brewn.

3. Work flow management engine:

In 1990 Business Process Management (BPM) was an interesting topic. In 2001 BPM came back. *“The principle of BPM is to provide business and technical users with a common framework to model, implement, deploy, execute, measure and improve business processes”*. [9]

There exist many work flow and BPM products in the market but all are expensive; JBOSS delivered a professional open source work flow/BPM engine called JBPM [10]. JBoss JBPM provides a high level view of applications and accomplishes several things:

- It facilitates more agile implementation of the processes required by business people.
- It describes business processes in a common dialect that lets business people and developers speak the same language.
- It organizes embedded logic of applications into separate and easily changed “state machines” to allow a new level of processes within businesses.

4. A reverse engineering module:

the main functionality of this module is to extract information from existing system to be used in modeling the integration with other systems as well as how this system well interact with other systems and how this system functionalities will be used.

5. Enterprise Service Bus (ESB):

“ESB is not a new software product — it's a new way of looking at how to integrate applications, coordinate resources and manipulate information” [11]. Unlike other approaches for connecting distributed applications, for example Remote Procedure Call (RPC), the ESB pattern enables the connection of software running in parallel on different platforms, written in different programming languages and using different programming models. ESB solves one of the current biggest challenges which is application integration.

the following are some of the existing ESB:

- Project Open ESB, Mule, Apache ServiceMix, Celtix, PetALS (JBI), IBM WebSphere Message Broker, IBM WebSphere ESB.

6. ESB UML meta model:

ESB UML meta model is a description that defines the structure of the ESB UML models as well as how the systems that are involved in the integration will talk to each other and to the ESB.

7. Transformation engine:

this engine will be responsible for:

- Generating Skeletal MDA models from the Business Process Models.
- Keeping the Business Process Models and the MDA Models in synchronization.
- and may generate the ESB xml configuration file depending on the way this engine will be implemented by as described in the following section.

9. An interface tool that integrates this components:

this will be a wizard tool for integrating the previous components.

3. Significance of the Study:

as indicated above **XBPF** needs frameworks, tools, modules, engines and integration between them. This study will not cover all of this areas but will concentrate on a specific areas. this research will extend the existing MDA framework to involve the business process model in its cycle. This will be achieved by taking the business process model and entering it to the transformation engine and MDA framework which will in turn generate the targeted system.

the previous goal will be achieved by one of the following ways:

1. Separation between the business model and the MDA model.
2. The same model will be used for both the business and MDA.

1. Separation between the business model and the MDA model:

in this way there will be two types of models Business Model and MDA Models. The transformation engine will generate MDA Models from the given Business Model Which will be processed by the MDA framework to produce the targeted system. (As in figure 3-1) following this way the transformation engine will not generate the ESB xml configuration file;

it will be generated by a custom andromda cartridge.

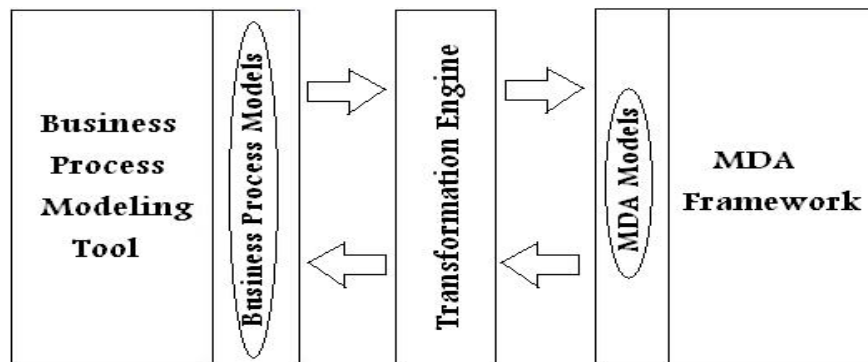


Figure 3-1

2. The same model will be used for both the business and MDA:

in this way the business model and the MDA model will be in the same model and there will be transformation (Business) cartridge in the MDA framework which will handle the transformation and the generation from the business part of this model. (As in figure 3-2) in this way the transformation engine (Business Cartridge) will generate the ESB xml configuration file.

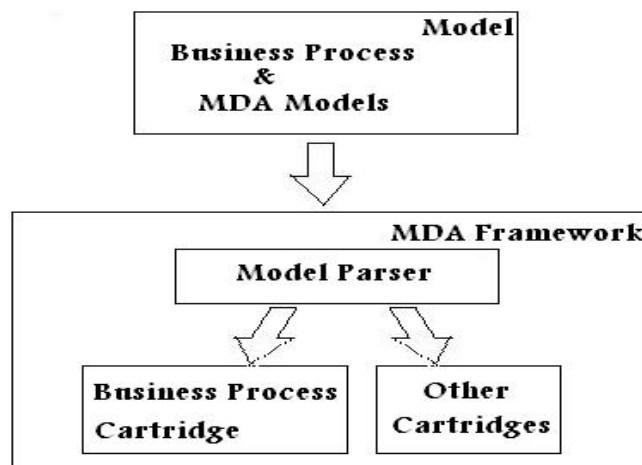


Figure 3-2

Following the first way there is expected problem, because at some point, Business Process models and MDA models of the system under development must interact with each other. Skeletal MDA Models could be generated from Business Process Models. Keeping the business process models and MDA models in synchronization over time would be a real problem, but this problem is the same problem that MDA frameworks faced in keeping Software models and generated code in synchronization.

4. Master Plan:

Action	Duration		Comments
	From	To	
1. Background	01/09/2006	31/10/2006	
2. Research	01/11/2006	31/01/2007	
3. Tool Development	01/02/2006	30/06/2006	
4. Tool Documentation	01/07/2007	31/07/2007	Include user guide, and developers guide.
5. Thesis Documentation	01/08/2007	31/08/2007	

5. References

- [1] MDA Guide Version 1.0.1.pdf - <http://www.omg.org>
- [2] ANDROMDA - <http://www.andromda.org>
- [3] 11-04 COL MDA - Frankel – Orr.pdf - <http://www.codegeneration.org>
- [4] BPMI - <http://www.BPMI.org>
- [5] BPML-2002.pdf <http://www.BPMI.org>
- [6] BPMN - <http://www.ebpmi.org/bpmn.htm>
- [7] BPEL4WS - <http://www.ebpmi.org/bpel4ws.htm>
- [8] MDA Guide Version 1.0.1.pdf - <http://www.omg.org>
- [9] BPMS Essentials with Open Source.pdf - <http://www.glintech.com>
- [10] JBOSS jBPM.pdf - <http://www.riseforth.com>
- [11] <http://www.ibm.com>