Development of Business Management System in Model-Driven Approach and Building of Model-Driven-Enabled Development Environment

Ming-Jen Huang (210205)

School of Information Science,
Japan Advanced Institute of Science and Technology

August 13, 2004

Keywords: Model-Driven, Roles, Responsibility-Driven, Collaboration-Based, Enterprise Computing.

1 Introduction

Model-driven approach is a software developing method that proposes programming from higher abstraction level and the computerized model transformation. Current software developing frameworks applies languages that can only be understood by developers to develop software. Users can not understand them, thus the developing materials can not be validated by the users. This hampered smooth communication between the users and the developers. Although OMG proposed MDA and defined a set of specifications to develop software from higher abstraction level, their works do not solve this issue. There is a gap that a model-driven software developing framework for enterprise computing that use a language that can be understood by the users and use as the communicating tool between the users and the developers is needed.

2 Enterprise Computing Software Development Framework (ECSDF)

In this thesis, I have proposed a model-driven software developing framework for enterprise computing, which is called ECSDF. It has two characteristics. The first is it raising the programming abstraction level by allowing the user and the developers to use the same model to describe problem domain. The second is it applies computational model transformation to transform the higher-abstraction-level model to object model and source code.

2.1 Architecture

ECSDF contains four architectural parts, the Business Model, the object model generator, the source code generator, and the virtual machine. The Business Model allows users and developers to use the same language to communicate and to define the details of software projects. The creation of the Business Model is inspired by the characteristics of enterprise computing. The Business Model consists of four models, the domain model, the document/view model, the business activity model, and business rules. The object model generator and the source code generator use the Business Model to generate object model and source code respectively. Object model and source code generating mechanism are based on the ideas of roles, responsibilities, and collaborations of software objects. This thesis mainly deals with the object model generator.

2.2 Object Model Generation

The mechanism can be simplified to "each system responsibility is taken by a set of collaborative software objects and, in turn, each software object takes smaller responsibilities and collaborates with its neighboring software objects." There are three types of rules, interaction rules, responsibility rules, and architectural rules to generate object model from the Business Model.

2.3 Development Environment for ECSDF

For ECSDF, a development environment, which is called ECSDF-DE, is built. It provides functionality of defining the business activity model and the document/view model and the rule-based object model generator. ECSDF-DE contains three architectural parts, the Eclipse plugin for graphical manipulation, the rule-based object model generator for generating object model, and data repository for the Business Model storage.

3 Evaluation of ECSDF

To evaluate the effectiveness of ECSDF, two business management systems are developed. One is a hand-coded, three-layered architecture software system (the hand-coded BMS for short), the other is an ECSDF-generated, collaboration-based software system (ECSDF-generated BMS for short). The evaluation is performed against the quality attributes of each software system. Although both BMS achieved their quality attributes, the ECSDF-generated BMS showed it has better productivity than the hand-coded BMS. The ECSDF-generated BMS is inferior in its flexibility when new technology emerges. The hand-coded BMS is inferior in maintainability because any change to the requirements will cause ripple effect due to the limitation of human's brain but not its design.

4 Conclusion

In this thesis, I have proposed a new model-driven software development framework. It consists of a domain-specific language, the Business Model, for the users and the developers to describe problem domain, an object model generator that based on the ideas of roles, responsibilities, and collaborations to transform the Business Model to object model and source code which can generator flexible and reliable object model. A development environment for ECSDF is built as Eclipse plugin. And finally the evaluation of ECSDF showed the generated software is not only productivity, but also flexible and reliable.