Model Synchronization for Efficient Software Application Maintenance

Terence C. Lau, Tack Tong, Ross Mckegney
Centre for Advanced Studies, IBM® Toronto Laboratory
{Lautc, Tacktong, rmckegne}@ca.ibm.com

Kostas Kontogiannis, Igor Ivkovic, Philip Liew
Dept of Electrical & Computer Engineering, University of Waterloo
{kostas, iivkovic, pliew}@swen.uwaterloo.ca

Ying Zou, Qi Zhang, Maokeng Hung
Dept of Electrical & Computer Engineering Queen's University
{ying.zou, qi.zhang, maokeng.hung}@ece.queensu.ca

Abstract

A business software application often has two perspectives: the business flow and operation that the application intends to solve, and the source code and its design and flow upon which the application is built. As one changes and evolves, the other needs to be synchronized. This report addresses an approach to synchronize these two models for effective software maintenance.

1. The challenge

Software applications and products are subject to constant changes. Customers modify the applications they acquire to suit their lines of business, and will continue to make changes as their business evolves, and as they move their systems from one generation of technology to the next. Software vendors make changes to their products as new functions are added in, or as design faults and programming errors are corrected. Behind this are, in fact, two groups of "stakeholders" and two perspectives of a software application: the *business users* whose primary perspective is the business tasks and their day-to-day operation and flow; and the *technical staff* whose focus is on the software information system, and the design, programming, and execution of its components.

There is then a need to help each of these two groups of stakeholders to maintain the application using their own model, and as changes are made, to keep the two models synchronized. Koehler and Di Lucca have previously discussed their work in this area. [1,2] We report on a software engineering methodology that makes the development and maintenance of software applications efficient and productive in such an environment.

2. The methodology

The methodology comprises three major aspects.

(1) Modeling the customer's business. The business model includes the workflow from a business operation perspective, its tasks, data, decisions, and people of different roles and authorities. We define a domain model and an XML representation of this business model, for annotation and processing purposes.

(2) Modeling the software information system. This includes a scanner that can analyze the source code of the software application for its constituent commands, tasks, logic flow, and database access, as well as a domain model and a graphical representation of the program flow in XML format.

(3) Synchronizing the two models. A tool is built to correlate the two models. This involves defining the ontology for the relationship between the two models, algorithm and facility to identify the relationship between two actual model instances, to track changes in each of them, and to keep them synchronized.

3. The report

We apply this approach on an integrated environment comprising the IBM® WebSphere® Business Integrator as the business modeling tool, the IBM WebSphere Commerce as the software information system, and the Eclipse Integrated Development Environment as the software development and maintenance system.

We report on our experience in defining the domain modeling, and in devising algorithms to analyze business operation and source code, from which domain models and flows are extracted, and the relationship between the two models are mapped and their changes tracked.

References

[1] J. Koehler, *et. al*. "From Business Process Model to Consistent Implementation: A Case for Formal Verification Methods". *Proceedings of 6th International Enterprise Distributed Object Computing Conference.* Lausanne, Switzerland, *2002.*

[2] G. Di Lucca, *et.al*. "Abstracting Business Level UML Diagrams from Web Applications". *Proceedings of 5th International Workshop on Web Site Evolution.* Amsterdam, the Netherlands, 2003.

The paper intends to represent the views of the authors rather than IBM. IBM and WebSphere are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04)
1063-6773/04 $20.00 © 2004 **IEEE**

IEEE COMPUTER SOCIETY