Chapter V

Enhancing Software Maintainability by Unifying and Integrating Standards

William C. Chu
Tunghai University, Taiwan

Chih-Hung Chang, Chih-Wei Lu, and Yeh-Ching Chung
Feng Chia University, Taiwan

Hongji Yang and Bing Qiao
De Montfort University, England

Hewijin Christine Jiau
National Cheng Kung University, Taiwan

Software standards are highly recommended because they promise faster and more efficient ways for software development with proven techniques and standard notations. Designers who adopt standards like UML and design patterns to construct models and designs in the processes of development suffer from a lack of communication and integration of various models and designs. Also, the problem of implicit inconsistency caused by making changes to components of the models and designs will significantly increase the cost and error for the process of maintenance. In this chapter, an XML-based unified model is proposed to help to solve the problems and to improve both software development and maintenance through unification and integration.
INTRODUCTION

Software systems need to be fast time-to-market, evolitional, interoperable, reusable, cross-platform, and much more. Maintaining software systems is now facing more challenges than ever before, due to 1) the rapid changes of hardware platforms, such as PDA, WAP phone, Information Appliance, etc., 2) emerging software technologies, such as Object Oriented, Java, middleware, groupware, etc., and 3) new services, such as E-commerce, mobile commerce, services for Application Service Provider (ASP), services for Internet Content Provider (ICP), etc.

Due to the high complexity of software systems, development and maintenance usually involve teamwork and high cost. However, most systems are developed in an ad hoc manner with very limited standard enforcement, which makes the software maintenance very difficult. De facto standards, such as Unified Modeling Language (UML) (OMG, 2001), or XML (Connolly, 2001; Lear, 1999), are used to reduce communication expenses during the software life cycle and to increase maintainability and reusability. Design Patterns (Gamma, Helm, Johnson, & Vlissides, 1995) are reusable solutions to recurring problems that occur during software development (Booch, 1991; Chu, Lu, Yang, & He, 2000; Holland, 1993; Johnson & Foote, 1988; Lano & Malik, 1997; Meyer, 1990; Ossher, Kaplan, Harrison, Katz, & Kruskal, 1995; Paul & Prakash, 1994; Xiao, 1994). However, these standards usually only cover partial phases of the software process. For example, UML provides standard notation for modeling software analysis and design. But lacking of support in the implementation and maintenance phases, design patterns offer help to the design phase, and component-based technologies focus on the implementation phase.

In other words, these standards are not talking to each other currently, and therefore designers need to spend a lot of manual effort mapping and integrating these standards while crossing each phase of the software life cycle. The activities of software maintenance involve the whole software life cycle, including requirement, design, implementation, testing, and maintenance phases. Not only the model used in each phase, but also the mapping and integration of models between phases will affect the efficiency of software maintenance.

Without unifying and integrating these standards, the consistency of the models cannot be maintained, and the extent of automation is very narrow. This chapter proposes an XML-based meta-model to unify and integrate these well-accepted standards in order to improve maintainability of the software systems.

This chapter will discuss the adopted standards, including UML, design patterns (Gamma et al., 1995), component-based framework, and XML. A comparison and mapping of these standards will be presented. An XML-based unified model is used to unify and integrate these various models.
Related Content

Coordination Languages and Models for Open Distributed Systems
[www.igi-global.com/article/coordination-languages-models-open-distributed/77614?camid=4v1a](www.igi-global.com/article/coordination-languages-models-open-distributed/77614?camid=4v1a)

Web-Wide Application Customization: The Case of Mashups
[www.igi-global.com/article/web-wide-application-customization/61394?camid=4v1a](www.igi-global.com/article/web-wide-application-customization/61394?camid=4v1a)

Proposals of a Method Detecting Learners' Difficult Points in Object Modeling Exercises and a Tool to Support the Method
Migrating Legacy System to the Web: A Business Process Reengineering Oriented Approach
www.igi-global.com/chapter/migrating-legacy-system-web/4902?camid=4v1a