Chapter IX

Developing a Software Testing Ontology in UML for a Software Growth Environment of Web-Based Applications

Hong Zhu, Oxford Brookes University, UK
Qingning Huo, Lanware, Ltd., UK

Abstract

This chapter introduces the concept of software growth environments to support sustainable long-term evolution of Web-based application systems. A multiagent prototype system is designed and implemented with emphasis on software testing. In this environment, software tools are agents that cooperate effectively with each other and human testers through communications at a high level of abstraction. New tools can be integrated into the system with maximal flexibility. These are achieved through the design and utilisation of an ontology of software testing that represents the knowledge of software engineering and codifies the knowledge for computer processing.
as the contents of an agent communication language. The ontology is represented in UML (Unified Modeling Language) at a high level of abstraction so that it can be validated by human experts. It is also codified in XML (Extensible Markup Language) for computer processing to achieve the required flexibility and extendibility.

**Introduction**

The Internet and Web are becoming a distributed, heterogeneous, and hypermedia computation platform, which stimulates many new progresses in software applications, see, e.g., Crowder, Wills and Hall (1998). However, Web-based applications are complex and difficult to develop and maintain. In Zhu et al. (2000), we argued that most Web-based applications are by nature evolutionary, and we proposed a growth model of software process. To support the sustainable evolutionary development of Web-based systems, we designed a multiagent architecture of software development and maintenance environment, and developed a prototype system for testing Web-based applications. A key feature of the architecture and the prototype system is the use of an ontology of software testing to facilitate the communications between agents, human developers and testers. In this chapter, we report the development of the ontology of software testing and its representation in UML.

The remainder of the chapter is organised as follows. The next section, Background and Motivations, gives the motivation of our research and briefly outlines our approach to the development and maintenance of Web-based applications. The structure and features of the multiagent software environment are described. A prototype system for testing Web-based applications is presented. Then the section on, Ontology of Software testing, reports the ontology of software testing and its representation in UML. Then we discuss the uses of the ontology in the prototype systems. The final section concludes the chapter with a discussion of related works and directions for future research.

**Background and Motivations**

**Characteristics of Web-Based Applications**

According to Lehman and Ramil (2001), software systems can be classified into three types according to what correctness means to the system. An S-type program is required to satisfy a prestated specification. For such a system, correctness is the absolute relationship between the specification and the program. A P-type program is required to form an acceptable solution to a stated problem in the real world. The correctness of a P-type program is determined by the acceptability of the solution to the stated problem. An E-type program is required to solve a problem or implement an application in a real-
Related Content

Specification and Checking of Dependency Relations between UML Models
[www.igi-global.com/chapter/specification-checking-dependency-relations-between/30544?camid=4v1a](www.igi-global.com/chapter/specification-checking-dependency-relations-between/30544?camid=4v1a)

Info-Mathics - The Mathematical Modeling of Information Systems
[www.igi-global.com/chapter/info-mathics-mathematical-modeling-information/30545?camid=4v1a](www.igi-global.com/chapter/info-mathics-mathematical-modeling-information/30545?camid=4v1a)

Keyword Search on XML Data
[www.igi-global.com/chapter/keyword-search-xml-data/41503?camid=4v1a](www.igi-global.com/chapter/keyword-search-xml-data/41503?camid=4v1a)

Automated Interpretation of Key Performance Indicators by Using Rules
[www.igi-global.com/chapter/automated-interpretation-key-performance-indicators/35877?camid=4v1a](www.igi-global.com/chapter/automated-interpretation-key-performance-indicators/35877?camid=4v1a)