A Multi-Approach Testing Methodology in TTCN for Web Applications

Pulei Xiong, Robert L. Probert, University of Ottawa, Ontario, Canada

Abstract—**Testing Web Applications (WAs) is challenging because it involves so many test methods and tools. In this fast abstract, we propose a multi-approach testing methodology for WAs, which enhances the testing of business rules of WAs, and integrates different testing approaches by specifying Test Suites in TTCN-3 on the abstract level. This helps separate test design from test implementation, increase the reusability of Test Cases (TCs), and increase the degree of automated testing.**

Index Terms—**Web Application, Software Testing, Integration, TTCN-3**

I. INTRODUCTION

Web Applications are becoming increasingly popular in the IT industry. Meanwhile, WAs has become more mission critical and complex; and utilizing new techniques, such as component techniques, XML, Web Services, etc.

WAs are thin-client Client/Server (C/S) systems. However, WAs are different from traditional C/S systems, comprising navigation-based interaction structure, n-tiered application architecture, heterogeneous, concurrent, and platform-independent aspects. WAs are also different from Web sites, since WAs always involve business logic implementation.

WA design and test are complex and require the integration of different methods. Not very much research has been conducted on modeling and testing of WAs. In this note, we propose a multi-approach testing methodology in TTCN-3 [9] for WAs. In the multi-approach, we build models for a WA on the analysis and design level with Extended UML notations. Then we apply multiple testing methods, such as grey box [8], black box, and white box, to test the WA based on the models and implementation. We integrate all these methods by specifying derived Test Cases on an abstract level using TTCN-3.

II. MODELING WEB APPLICATIONS

WAs are usually based on n-tier architecture. Such applications may be configured as a *client tier* to provide the user interface, a *web server tier* to provide the presentation of a user interface, one or more *middle-tier modules* to provide business logic for an application, and back-end enterprise information systems (EISs) providing data management. Many techniques are employed to support this n-tier architecture. For example, JavaScript, VBScript, Java Applet, and ActiveX controls are used on the client tier; CGI, NSAPI/ISAPI, ASP, JSP, PHP, and Java Servlet are used on the web server tier; EJB, ActiveX control, JDBC, and ADO are used on the middle-tier; SQL and other database products are used for back-end EIS. WAs are so complex that it is necessary to model them.

UML, an OMG standard modeling language, is suitable to model WAs. Some models have been proposed to represent a WA with extended UML [1 2 3]. [1] extends UML to represent web-specific components. [2] improves the extended UML proposed in [1] to build a relatively complete set of models on the design level. [3] contributes to address navigation issues by applying UML to requirements analysis. In addition, UML can also be used to model the EIS tier.

III. TESTING WEB APPLICATIONS

A. Related work

A few methods, based on models, implementation, or user profiles, have been proposed for testing WAs. [4] proposes an Object-based testing method based on a 3-tier WA architecture. [5] proposes a test approach that is based on models of Object Relation Diagrams (ORD), Page Navigation Diagrams (PND), and Object State Diagrams (OSD). In addition, [5] also presents an Object-based data flow method discussed in detail in [6], which employs the Block Branch Diagrams (BBD) and Function Cluster Diagrams (FCD). [7] proposes an approach based on high level design model and code analysis, with its proprietary tools support.

However, there are some limitations in these testing methods. For example, most of these approaches tend to overlook testing the Middle tier, which implements the business logic, the most important part in a WA.

B. A multi-approach testing methodology for WAs

WA test involves many types of testing, such as GUI testing, Functional testing, Database testing etc [8]. In order to implement this testing in an integrated environment, we propose a multi-approach testing methodology for WAs, which is based on multiple models, implementation, and user profiles; applies multiple test methods, such as grey box, black box, white box; integrates specification of TCs on the abstract level using TTCN-3; chooses different tools to test different aspects of WAs; and focuses on unit and integration levels according to the n-tier architecture of WAs.

The components of our testing methodology are (see Figure 1):

1) The *models* we employ include *analysis model*, such as use case, object state diagram; and *design model*, such as *navigation model*, *business model*, *data model*, sequence diagrams etc.

2) Multiple test methods are employed for different models.

3) TTCN-3 is used to specify all the TCs on an abstract level (ATS). This helps: (1) separate test design from test

implementation; (2) make ATS language and platform independent; (3) increase the reusability of TCs; (4) increase the degree of automated testing; (5) take advantages of the commercial tool support enjoyed by TTCN-3.

4) Several test tools may be used to facilitate aspects of WA testing [8], such as link checkers, and GUI Capture and Playback tools. Choose appropriate tools to test a specific aspect of WAs.

5) TTCN-3 translators may be required to convert ATS to the proprietary scripting languages of test tools.

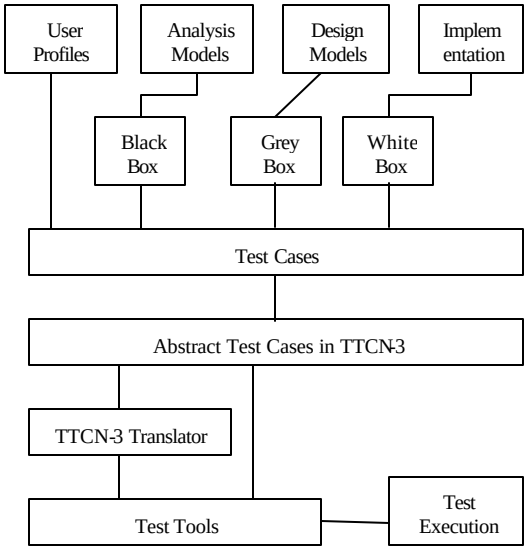

Figure 1 A multi-approach testing methodology conceptual architecture

Our proposed testing approach comprises two phrases: *unit testing* and *integration testing*, according to the n-tier architecture of WAs. Unit testing includes database testing, business logic testing, and navigation testing.

Database testing is based on the data model. Database extensions, such as stored procedures, triggers, and rules, are usually used to increase database functionality and performance. Grey box testing methods can be employed for testing functional aspect of the database on the unit level.

Business logic testing is based on models for middle tier. Middle tier is a typical Object-oriented software system. [10] presents several grey- and white-box testing methods, e.g. state-based and constraint-based testing techniques, to implement class level testing, interaction testing, and inheritance testing. Models utilized here may include class diagrams with constraint in OCL, class state transition diagrams, etc.

Navigation testing is based on models, e.g. navigation model, and web page source codes. Both grey- and white-box methods introduced in [5 6] are useful.

Integration testing follows unit testing. In this phrase of testing, web page collaborations are taken into account. Black box methods are employed. Functional use case diagrams, navigational use case diagrams, and state evolution diagrams are used to derive test cases on the system level.

IV. SUMMARY AND FUTURE WORK

The purpose of proposing a multi-approach testing methodology is to build a comprehensive testing architecture and process for functional testing of WAs leading to derived TCs in TTCN-3. We will evaluate this approach by applying it to typical WAs, and developing some prototype tools for automating some of the test process.

REFERENCES

[1] Jim Conallen, *Modeling Web Application Architectures with UML*, Rational Software White Paper, June 1999

[2] Jingfeng Li, Jian Chen, and Ping Chen, *Modeling Web Application Architecture with UML*, 36th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-Asia'00), Nov. 2000

[3] Luciano Baresi, Franca Garzotto, and Paolo Paolini, *Extending UML for Modeling Web Applications*, 34th Hawaii International Conference on System Sciences, 2001

[4] Ji-Tzay Yang; Jiun-Long Huang; Feng-Jian Wang; Chu, W.C.; *An object-oriented architecture supporting Web application testing*, 23rd Annual International Computer Software and Applications Conference, 1999

[5] Kung, D.C.; Chien-Hung Liu; Pei Hsia; *An object-oriented web test model for testing Web applications*, First Asia-Pacific Conference on Quality Software, 2000

[6] Chien-Hung Liu; Kung, D.C.; Pei Hsia; Chih-Tung Hsu; *Structural testing of Web applications*, 11th International Symposium on Software Reliability Engineering, 2000

[7] Ricca, F.; Tonella, P.; Analysis and testing of web applications, 23rd International Conference on Software Engineering, 2001

[8] Hung Q. Nguyen, Bob Johnson, and Michael Hackett, *Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems*, Second Edition, John Wiley & Sons, Inc., 2003

[9] ETSI ES 201 873-1: *Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language*

[10] John D. McGregor and David A. Sykes, *A practical guide to testing Object-Oriented software*, Addison-Wesley 2001

Pulei Xiong is a Master student in the program of System Science, school of Management, University of Ottawa. He is working on Web Application and E-Commerce system testing under the supervision of Dr. Robert L. Probert.

His research interests include Web Application and E-Commerce system testing, formal verification and validation, automatic test case generation, and TTCN.

Robert L. Probert (Ph.D., Computer Science, University of Waterloo, 1973) is Full Professor and was the founding Director of the School of Information Technology and Engineering (SITE) and Coordinator of the Nortel Networks ASERT Lab at the University of Ottawa. He has been a Visiting Scientist with IBM CAS since 1998, and is now a Faculty Fellow in E-Commerce Testing. His research interests and publications are primarily in testing protocols and communications software quality engineering.