The First International Workshop on Automation of Software Test

Hong Zhu
Department of Computing
Oxford Brookes University
Oxford OX33 1HX
United Kingdom
Email: hzhu@brookes.ac.uk
Tel: +44 1865 484580
Fax: +44 1865 484545

Joseph R. Horgan
Telcordia Technologies
One Telcordia Drive,
RRC-1M322, Piscataway,
NJ 08854, USA
Email:jrh@research.telcordia.com
Tel: 732-699-2580
Fax: 732-336-7015

S.C. Cheung
Department of Computer Science
The Hong Kong University of
Science and Technology
Clear Water Bay, Hong Kong
Email: sccheung@cs.ust.hk
Tel: +852 2358-7016
Fax: +852 2358-1477

J. Jenny Li
Avaya Labs
233 Mount Airy Road
Basking Ridge, NJ 07920, USA
Email:
jjli@research.avayalabs.com
Tel: 908-696-5147
Fax: 908-696-5402

Categories and Subject Descriptors
D. [**Software**]; D.2.0 [**Software Engineering**]: General – *methodologies, techniques, tools*;
D.2.5 **[Testing and Debugging]**: *Testing tools*.

General Terms
Design, Reliability, Security, Experimentation, Languages, Management, Measurement.

Keywords
Software test, Software automation, Test case generation, Model-based test, Test adequacy and coverage, Test cost and effectiveness, Test tools and environments, Component integration test.

1 MOTIVATION

Software testing is indispensable for all software development, but labor intensive and expensive. In software development practice, testing can account for over half of the total development efforts. It is imperative to reduce the cost and improve the effectiveness of software testing by automating the testing process, which contains many testing related activities using various techniques and methods.

Automation is the trend of software testing. In the past decades, a great amount of research effort has been made on the automation of test case generation, test oracles and so on. However, the current practice of software test automation is mostly based on recording manual testing activities and replaying recorded test scripts for regression testing. Bridging the gap between theory and practice will not only significantly improve the current-state of software production, but also foster innovative research in the area. As the theories of software testing become more mature, a larger scale automation of the testing process becomes feasible. Indeed, many software test tools have been made available on the market in the past few years. However, few of them have taken inter-operability into serious consideration. Software systems have become more and more complicated with components developed by different vendors and using different techniques in different programming languages and even run on different platforms. Few software testing tools can support all testing tasks within one tool. Therefore, it is timely and important for the development of software testing methodologies into a scientific discipline as a part of software engineering. The workshop aims at providing researchers and practitioners a forum for exchanging ideas, experiences, understanding of the problems, visions for the future, and promising solutions to the problems. The workshop also serves as a platform for researchers and developers of testing tools to work together to identify the problems in the theory and practice of software test automation and to set an agenda and lay the foundation for future development.

2 THEME AND TOPIC

The theme of the workshop focuses on bridging the gap between the theories and practice of software test automation. Its related topics can be characterised by the following five aspects.

1) Methodology. This aspect is concerned with the research and practices of software test automation in the context of various software development methodologies, such as traditional heavy weight methodologies, rapid prototyping and evolutionary development methodology, component-based software development, object-oriented software development, agile and test-driven methodology, software architecture and product lines, and service-oriented software engineering, etc.

2) Technology. This aspect covers issues related to the automation of various test techniques and their corresponding components used in various test related activities, such as test case generation, test oracle and test result checking, test driver, stubs, harness and test script generation, software instrumentation, test adequacy and coverage measurement, and in general the generation of any test related software artifact, the management of testing activities and recourses, etc. It also includes the underlying techniques that support various software testing methods, such as structural testing, functional testing, error based testing, fault-based testing, partition testing and combinatorial testing, random testing, program-based testing, specification-based testing, model-based testing, risk-based testing, etc.; just to mention a few.

3) Applications. It is concerned with the development and application of automation methodologies and techniques in the testing of various specific types of software in various application domains, such as Internet and Web-based applications, Web Services, peer-to-peer applications and Grid systems, Semantic Web, database applications and information systems, systems software such as middleware, architecture and reference models, XML schemes, compilers, OS, real-time systems, concurrent and parallel systems, communication systems and protocols, embedded systems, etc. Besides application domains, it will consider issues related to application of testing automation, such as cost, scalability, economical impact, etc.

4) Tools and environments. It is devoted to the issues in the development, operation, maintenance and evolution of software testing tools and environments, such as the functional, architectural and interface designs of automated software testing tools and environments, the construction and evaluation of practical and prototype systems of automated testing and implementation issues, and the survey and comparative study of existing test automation tools.

5) Experiments, empirical studies, experiences and vision of the future. This aspect is concerned with the experiments and empirical studies of the impact of software test automation on software development practice, and reports on real experiences using automated testing techniques, methods and tools in industrial settings, such as the effectiveness of automated testing tools, methods and techniques, such as fault detecting abilities, the cost of building and using the automation versus savings from the automation, the usability of various techniques, methods and tools. An important part of the issues is the identification of problems that hamper the wider adaptation of automated test techniques, methods and tools as well as the analysis and specification of the requirements on automated software testing.

The workshop received a good number of submissions of papers authored by a total of 90 academic researchers and industrial practitioners from 15 different countries. The submitted papers cover a wide range of topics in the area of software test automation. Most papers are concerned with more than one of the above five aspects. The following table shows the distribution of the topics covered by the submitted papers according to the keywords of the papers.

Table 1. Distribution of Concerned Aspects

Topic	**Percentage of Submissions**
Methodology	26.7%
Technology	83.3%
Application	36.0%
Tool and Environment	40.0%
Experiment, Practice and Empirical study	20.0%

As shown in the above table, more than 83% of submitted papers are concerned with the technological aspect of software test automation. Table 2 below shows the distribution of these papers over the specific issues of software test technologies, where many papers addressed more than one technical issue.

Table 2. Distribution of the concerned technical issues

Topic	**Number of submissions**
Test case generation	60%
Management of test activities and resources	32%
Test drive, harness and script	24%
Test adequacy and coverage measurement	16%
Test oracle	4%

The above statistical data show a high research interest on automatic test case generation from both academic and industrial authors. After reviews by the workshop program committee members, 18 papers of high quality were selected for presentation at the workshop and publication in the workshop proceedings.

3 PLANNED WORKSHOP PROGRAM

The workshop will include five sessions. Each session focuses on one specific topic.

Session one includes four papers on model-based automatic testing. Chen, Qiu, and Li present a method to generate tests from UML activity diagrams to satisfy certain coverage criteria. Vieira *et al*. describe an ongoing research on test case generation from UML diagrams based on an idea of combining data and graph coverage of UML models. Pfaller *et al*. present a method for combining testing coverage and system requirements. They discuss the issue of various levels of model abstractions in the context of embedded systems. Lund and Stolen use sequence diagrams with advance features such as negation and assertion not only for automatic test case generation but also for automatic checking of test results.

Session two has three papers of test generation based on program input domains. Liu and Tan address the issues of statistical input validation for input control. Ji *et al*. propose an approach to the generation of test cases that enables monitoring program execution time more precisely in test. Shan and Zhu propose a data mutation method to the generation of a large number of structurally complicated test data from a few seed test cases. The method has been applied to a modeling tool and proven effective.

Session three includes three papers on component and integration testing. Abdurazik and Offutt propose a solution to integration testing based on class coupling for testing orders. The paper by Gallagher and Offutt presents a tool for automated testing of inter-operating OO classes. The tool operates directly on an object-oriented software specification to produce a data flow graph and executable test cases that adequately cover the graph according to classical graph coverage criteria. Yuan and Xie describe a framework for automatic generation of integration tests based on call-sequence constraints inferred from dynamic executions.

Session four addresses economical issues of test automation with four papers. Cai *et al*. propose and validate their solution to test automation under cost constraints. Offutt *et al*. present the results of an empirical study on the affordability of class level mutation testing using MuJava tool. Ramler and Wolfmaier present their study of economical issues in testing automation and discuss how to balance between automated testing and manual testing to achieve an optimal test cost. Artho *et al*. focus on the scalability issue and extend unit testing framework for large scale tests.

The last session of the workshop, session five, is devoted to test tools and environments. It includes four papers. Yang, Li and Weiss survey and compare a set of coverage-based testing tools aiming at providing guidelines to the selection of coverage tools. Okika *et al*. describe a prototype TTCN-3 test harness for legacy software systems. Sauve *et al*. present a tool to create and execute client-readable acceptance tests in an acceptance-test driven software development process. Delamaro *et al*. describe a strategy for coverage-based testing of mobile devices on both emulators and the real mobile devices and report a test environment that supports the strategy.