International Journal of Innovative
Computing, Information and Control
Volume **7**, Number **6**, June **2011**

 ISSN 1349-4198
pp. **3281–3294**

FULLY AUTOMATED GUI TESTING AND COVERAGE ANALYSIS USING GENETIC ALGORITHMS

Abdul Rauf, Arfan Jaffar and Arshad Ali Shahid

Department of Computer Science
National University of Computer and Emerging Sciences
Islamabad, Pakistan
{ a.rauf; arfan.jaffer; arshad.ali }@nu.edu.pk

Received November 2009; revised March 2010

Abstract. *Graphical user interface (GUI), is a graphical front-end to a software system, contains graphical objects with certain distinct values whose purpose is to ascertain the state of the GUI at any time. In order to ensure that the quality of the software is par excellence, software developing organizations endeavor to test the software meticulously. Nevertheless, the process of testing a GUI application calls for a colossal effort, owing on account of the intricacy entailed in such applications. Subsequently, organizations were spurred to initiate the automation of GUI testing, thereby proposing various techniques to achieve this end. A GUI model event-flow graph, an innovative technique being utilized in the field of automated GUI testing, represents, likewise control flow graph, all promising progressions of events that can be executed on GUI. The search for utmost quality insurance for software, through the introduction of automated software testing, raises yet another challenging question, that of the "amount" of testing required so as procuring the best results. In the course of the development of the techniques for the automation of the software testing procedure, a few measures can be employed to provide guidance on the quality of an automatic test suite. Based on some predefined test criterion, genetic algorithm searches for the best possible test parameter combinations. Usually, this test criterion corresponds to a "coverage function" that measures how much of the automatically generated optimization parameters satisfies the given test criterion. In this paper, we have attempted to exploit the event driven nature of GUI. Based on this nature, we have presented a GUI testing and coverage analysis technique centered on genetic algorithms.*

Keywords: GUI testing, Genetic algorithm, Coverage criterion, Coverage analysis, Event flow, Test data generation, Test path, Automation testing

1. **Introduction.** Quality of the delivered software depends heavily on the systematic activity of software testing. Testing related activities go on with the entire development life cycle and may consume a large fraction of the effort required for producing software [1]. Purpose of software testing is to improve software quality and increases confidence in software's proper functioning. This purpose is achieved with support of software testing activities [2]; these activities include collecting test data, generation and execution of test cases, filtration and reduction of test cases, coverage analysis and reporting. Software testing is a labor intensive process and studies indicate that more than 50Graphical user interfaces (GUIs) are one of the most important components of modern day software and are being considered as necessary part for most of today's software. GUIs give user a relatively more ease and freedom to interact while accessing with the system [3]. Recognizing the importance of GUIs, software developers are dedicating more effort, up to 50Common practice of GUI test designers is to generate and execute test cases to traverse parts of GUI application. These test cases need to focus on a subspace in order