Testing applications involves the creation of test sets, their execution, management and adequacy assessment. An extensible framework, called RiOT, has been developed to allow the implementation of tools for test coverage measurement, test execution management and fault-based testing of distributed Java applications. Test coverage is measured in terms of elements covered in the interface description of an application’s components. Mutation operators for interface mutation analysis are proposed and their implementation described. Fault injection testing for the assessment of the application’s fault tolerance properties is made possible using the fault injection mechanism of the framework. An extension of the framework to allow test monitoring and control is described. The framework uses a hierarchical form of data communication between its different modules. The framework is designed to be scalable and easily extensible for future enhancements.

Keywords: Distributed applications, fault injection testing, Java RMI, Jiro, mutation testing, software testing, test adequacy criteria, test coverage, test management.

INTRODUCTION

Current software applications are highly complex. They are usually large-scale, distributed and heterogeneous, and have complex requirements. These factors increase the complexity of testing such applications. The state-of-the-art in software testing has not been able to keep pace with the developments in new distributed application technologies.

As part of a general testing process, a tester needs to develop test cases, execute them and observe failures. If there are failures, the cause is determined and necessary corrections are made. Given the infinite domain of inputs to an application, testing can never guarantee that the software is error free. At a practical level, testing is limited by the amount of resources available. Testing is only as good as the degree to which it “exercises” an application. A test suite that executes only a portion of an application with respect to a certain coverage domain, cannot be termed adequate with respect to that coverage domain. The effectiveness of testing performed is measured using a coverage metric (Zhu, Hall and May, 1997).

A test set is defined to be adequate with respect to some coverage metric if the test cases cover all the elements in the domain for that particular coverage metric. For example, if a test set executes