Chapter 7.17
A Metamorphic Testing Approach for Online Testing of Service-Oriented Software Applications

W. K. Chan
Hong Kong University of Science and Technology, Hong Kong

S. C. Cheung
Hong Kong University of Science and Technology, Hong Kong

Karl R. P. H. Leung
Hong Kong Institute of Vocational Education, Hong Kong

ABSTRACT

Testing the correctness of services assures the functional quality of service-oriented applications. A service-oriented application may bind dynamically to its supportive services. For the same service interface, the supportive services may behave differently. A service may also need to realize a business strategy, like best pricing, relative to the behavior of its counterparts and the dynamic market situations. Many existing works ignore these issues to address the problem of identifying failures from test results. This article proposes a metamorphic approach for online services testing. The off-line testing determines a set of successful test cases to construct their corresponding follow-up test cases for the online testing. These test cases will be executed by metamorphic services that encapsulate the services under test as well as the implementations of metamorphic relations. Thus, any failure revealed by the metamorphic testing approach

will be due to the failures in the online testing mode. An experiment is included.

INTRODUCTION

The service-oriented architecture (SOA) is an architectural reference model (Bass et al., 2003) for a kind of distributed computing such as the Web services (W3C, 2002). It promises to alleviate the problems related to the integration of heterogeneous applications (Kreger et al., 2003; Mukhi et al., 2004). In this reference model, a SOA application is denoted by a collection of self-contained communicating components, known as *services*. The model also emphasizes that each service should make little or no assumption about its collaborating services. This setting advocates the dynamic composition of a service by using different configurations of supportive services, creating behavioral differences amongst different invocations of a service. Typical end-users of a SOA application, such as bank customers using an online foreign exchange trading service, may expect consistent outcomes each time they use the service. The customers may further compare the online foreign exchange service of a bank to similar services of other banks to judge whether the service is of good quality. If a B2B service provider is driven by a predefined business strategy to, for example, maintain its market share, then the criteria to define functional correctness of the service may vary according to its environment. In other words, testers need to integrate the environment of a service to check test results.

Services may be subject to both the off-line testing and the online testing. Unlike the testing of conventional programs, services bind dynamically to other peer services when it is tested online. While the off-line testing of services is analogous to the testing of conventional programs, the online testing of services needs to address new issues and difficulties.

Testers may generally apply certain static analyses or testing techniques to assure the correctness of a software application. The evaluation criteria of the functional correctness, on the other hand, must be predefined. For a unit testing, testers also want to address the test case selection problem and the test oracle problem (Beizer, 1990). We restrict our attention to the latter problem in this article.

A test oracle is a mechanism that reliably decides whether a test succeeds. For services, as we will discuss, formal test oracle may be unavailable. The expected behavior of a service that represents business goods and services changes according to the environment. Such an expected behavior is relative to the behaviors of competing services or other services. Intuitively, it is hard to define the expected behavior explicitly in the first place. Tsai et al., (2004) for example, suggest using a progressive ranking of similar implementations of a service description to alleviate the test oracle problem. The behaviors of different implementations of the same service vary in general. Test results of a particular group of implementations cannot reliably be served as the expected behavior of a particular implementation of the same service on the same test case. Also, a typical SOA application may comprise collaborative services of multiple organizations, knowing all the implementations are impractical (Ye et al., 2006). For example, a travel package consultant may wrap up services of various hotels, airlines, and entertainment centers to personalize tour packages for a particular client. Without the implementation details, static analysis appears infeasible to assure the implementation quality. The black box approach for test results checking remains viable and popular to assure the correctness of a software application.

Metamorphic testing is a promising approach to the test oracle problem in the testing of conventional scientific programs (Chan et al., 1998; Chen et al., 1998, 2002). Instead of relating an