ABSTRACT

Many web services not only communicate through XML-based messages, but also may dynamically modify their behaviors by applying different interpretations on XML messages through updating the associated XML Schemas or XML-based interface specifications. Such artifacts are usually complex, allowing XML-based messages conforming to these specifications structurally complex. Testing should cost-effectively cover all scenarios. Test case prioritization is a dimension of regression testing that assures a program from unintended modifications by reordering the test cases within a test suite. However, many existing test case prioritization techniques for regression testing treat test cases of different complexity generically. In this paper, the authors exploit the insights on the structural similarity of XML-based artifacts between test cases in both static and dynamic dimensions, and propose a family of test case prioritization techniques that selects pairs of test case without replacement in turn. To the best of their knowledge, it is the first test case prioritization proposal that selects test case pairs for prioritization. The authors validate their techniques by a suite of benchmarks. The empirical results show that when incorporating all dimensions, some members of our technique family can be more effective than conventional coverage-based techniques.

Keywords: Adaption, Extensible Markup Language (XML) Similarity, Pairwise Selection, Service Evolution, Test Case Prioritization

INTRODUCTION

A WS-BPEL web service (WS-BPEL Version 2.0, 2007) may interact with other web services that collectively implement a function. Any maintenance or runtime adaptation of the web service may potentially result in faults or cause incompatible interactions between this web service and its belonging composite services. To validate whether an evolved version of the web service conforms to its previously established functional behaviors, a testing agent (which
can be a web service) may apply a test suite to check whether the evolved version of the web service correctly handles the test suite.

A WS-BPEL web service consists of a WSDL document and a workflow process. A WSDL document (W3C WSDL 1.1, 2001) is an XML-based document. It contains the specifications that define both web service descriptions and message data types specified in XML schemas (Fu, et al., 2004). A web service description provides a locator of a web service, and represents a web service as a collection of operations. Individual workflow steps in a workflow process may apply different specifications defined in WSDL (dubbed WSDL specifications) to different portions of the same or different XML documents. Workflow programs (e.g., a BPEL program) are popular to be encoded in XML form as well (BPEL Repository, 2006; Oracle BPEL Process Manager 10.1.2, 2007; WS-BPEL Version 2.0, 2007).

However, two XML-based documents sharing the same set of tags may structure these tags in quite different ways, potentially causing the same web service to produce radically different results for the two messages (Mei et al., 2011). An XML document can be modeled as an ordered, labeled tree (Garofalakis et al., 2005; Guha, et al., 2002; Mei et al., 2011). A mathematical property of such a tree is that a given tree must subsume every sub-tree or any resultant tree generated by removing any sub-trees from the given tree. We thus observe that the element coverage with respect to an XML document may not help reveal the coverage characteristics in relation to the internal structure of an XML document.

A regression testing technique that does not consider these characteristics can be less effective to assure WS-BPEL web service. For instance, an online hotel booking application may use the booking service provided by a hotel to handle every room booking request. Without further manipulation of a message, this application may relay the booking result returned by the hotel back to the client service. Multiple program executions may simply exercise the same workflow steps, irrespective to the contents of the returned XML messages (e.g., indicating successful booking or not), which may require different program logics to handle them.

This paper proposes a suite of similarity-based test case prioritization techniques for the regression testing of WS-BPEL web services based on the pairwise selection strategy. Pairwise comparison is a fundamental strategy to examine elements and pick associations between elements in a finite set. To the best of our knowledge, no existing test case prioritization techniques formulated directly on top of this type of strategy has been proposed for the regression testing of web services. Our techniques compute the structural similarity of XML-based artifacts between test cases. They progressively consider the XML-based artifacts in three levels: WSDL interface specification, XML-based messages, and BPEL workflow process. Each technique assigns the execution priorities to the test cases in a regression test suite by assessing the similarity values of test case pairs. They select test pairs either iteratively or following a heuristic order. We have conducted an experiment to validate our techniques. The empirical results show that some of the proposed techniques can achieve higher rates of fault detection, in terms of APFD, than other studied techniques and random ordering when using all levels of XML-based artifacts.

The preliminary version (Mei et al., 2013) of this paper has presented the basic idea of our techniques. In this paper, we propose two new similarity-based techniques (M5 and M6) based on the iterative similarity strategy, elaborate the design rationale of all proposed test case prioritization techniques, and strengthen the experiment by conducting more data analysis including hypothesis testing to validate the observations.

The main contribution of this paper, together with its preliminary version, is twofold: (i) To the best of our knowledge, this paper is the first work that formulates pairwise test case prioritization techniques for the regression testing of web services. (ii) We report an experiment that validates the feasibility of our proposal.
Parallel Computing for Mining Association Rules in Distributed P2P Networks
www.igi-global.com/chapter/parallel-computing-mining-association-rules/53273?camid=4v1a

DiALog: A Distributed Model for Capturing Provenance and Auditing Information
www.igi-global.com/article/dialog-distributed-model-capturing-provenance/42107?camid=4v1a