Extending WSDL to Facilitate Web Services Testing

W. T. Tsai*, Ray Paul†, Yamin Wang‡, Chun Fan*, and Dong Wang*
*Dept. of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287
†Department of Defense, Washington DC
‡Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455

Abstract
Web services might be the most popular and powerful software development technology in today's software world. Yet it brings software developers and tester a lot of challenges also. This is mainly caused by the insufficient information provided by the WSDL file of a web service. From the WSDL file, we can not get the information useful for testing such as dependence information. To solve this problem, we proposed and practiced to extend the WSDL to support this kind of information description. In this paper, four kinds of extension: input-output dependency, invocation sequence, hierarchical functional description and concurrent sequence specifications. Also their advantages are discussed.
Keywords testing Web services WSDL

Introduction

Web services have received significant attention recently as many companies including Microsoft are pushing this technology. Web services have many attractive features such as using standard protocols including SOAP, WSDL and UDDI. These standard protocols address interoperability issues and they are developed for application development and runtime web service selection and invocation. Unfortunately, current WSDL does not contain sufficient information for an application engineer to test the available web services. Testing web services is equivalent to black-box testing where specifications are available but no design or code are available, and the specification is written in WSDL, while code can be written in Java, C#, or other programming languages. The current WSDL contains the following information: the number of inputs and outputs, the variable type of each input and output, the order the inputs are given and outputs are returned, and how a web service should be invoked.

For example, from the standard WSDL description file for a web service, we can only know the inputs, outputs, the types of the inputs and outputs, the order of the inputs and outputs, and how the web service should be invoked. The information such as input/output dependence between web services, and the invocation sequence can not be obtained from the WSDL file.

To perform black-box testing and regression testing, it is important to know more than those provided by WSDL. Specifically, the following information is needed:

- Input-output Dependency
- Invocation Sequences
- Hierarchical Functional Description
- Concurrent Sequence Specifications

Input-Output Dependency

In numerous testing strategies such as regression testing and data flow testing, it is important to know the input and output dependency. To do so, we introduce a new complex type in the WSDL schema, WSInputOutputDependenceType, which includes at least one element of type WSIOPairType. The WSIOPairType is also a complex type which has two sub-elements, Input and Output, of WSIOModeType. This information can be generated by dependence analysis inside a web service. The association between the inputs and outputs may help eliminate unnecessary test cases for carrying out the regression test. It also helps to produce expected outputs.

Invocation Sequences

A web service can have other web service to perform its tasks, and thus it is important to trace and state those calling relationship. MtSS (Method Sequence Specification) was developed to specify those calling relationships for OO programs [Kirani 1994] while IMOZU diagrams have been developed for procedural programs [Onoma 1996]. Similar information can be captured in WSDL dependence. To do this, we define a web service reference type, which has web service name and the link to its WSDL document: WSInvocationDependenceType to represent the callers and callees. There are two sub-elements in it, WSICallers and WSICallees of WSICallersType and WSICalleesType respectively. For both the WSICallersType and WSICalleesType, there is a link to their WSDL files for reference. By tracing this information among participating web services, it is

Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering (HASE'02)
1530-2059/02 $17.00 © 2002 IEEE

possible to generate the complete calling sequence, and the calling sequence is useful in path testing and data flow testing.

Hierarchical Functional Description

In addition to providing structural information such as dependency and calling sequences. It is possible to incorporate functional descriptions into WSDL. Furthermore functional descriptions can be organized in a hierarchical manner and can embedded in WSDL. Once organized in a hierarchical manner, a functional description can be form by sub-description. The WSDL can be extended to include two sub-elements, WSFParents and WSFChildren, which are of WSFParentType and WSFChildrenType. For both of these two types, there is a link to their corresponding WSDL files to reflect the hierarchical structure. The WSFParents is the main description, while the WSFChildren is the sub-description. By using this functional description, it is possible to provide analysis such as functional dependency analysis, and thus greatly improve the capability to perform various types of testing such as functional testing and regression testing. Also, by linking the functional description with an OO test framework [Tsai 2002a, Tsai 2002b], many of test execution can be automated.

Sequence Specifications

Sequence specifications have been found to be useful for testing OO programs [Kirani 1994, Wang 1996, Tsai 1999] as well as deriving test cases from formal specification languages such as Z [Tsai 1999]. Sequence specifications capture calling sequences in OO frameworks [Tsai 1999], concurrent behaviors [Wang 1996] and timing aspects for real-time systems [Vishnuvajalla 2001]. A web service can have similar sequence specifications in WSDL: Specifically, the following show how sequence specifications can be incorporated in WSDL: for instance, the following OpenAccount • Deposit • (Deposit | Withdraw)* • CloseAccount. We can have this WSDL description:

```
<WSSequenceSpecification xmlns="http://tempuri.org/"
Mode="Single">
    <WSSSeries Mode="Single">
        <WSRef>  <Name>OpenAccount</Name>
<RefLink>http://www.WebServiceExample.com/OpenAccount.wsdl</
RefLink>
        </WSRef>
    </WSSSeries>
    <WSSSeries Mode="Single">
        <WSRef>  <Name>Deposit</Name>
<RefLink>http://www.WebServiceExample.com/Deposit.wsdl</RefLin
k>      </WSRef>
    </WSSSeries>
     <WSSSeries  Mode="ZeroOrMore">
        <WSSMutualExclusive Mode="Single">
            <WSRef>  <Name>Deposit</Name>
<RefLink>http://www.WebServiceExample.com/Deposit.wsdl</RefLin
k>    </WSRef>
        </WSSMutualExclusive>
        <WSSMutualExclusive Mode="Single">
            <WSRef> <Name>Withdraw</Name>
<RefLink>http://www.WebServiceExample.com/Withdraw.wsdl</RefL
ink>  </WSRef>
        </WSSMutualExclusive>
    </WSSSeries >
    <WSSSeries Mode="Single">
        <WSRef> <Name>CloseAccount</Name>
            <RefLink>http://www.WebServiceExample.com/
CloseAccount.wsdl</RefLink>    </WSRef>
    </WSSSeries>
</WSSequenceSpecification>
```

These extensions have been used in testing applications using web services on the .NET platforms. By using these extensions, we can easily retrieve the necessary useful information for web service testing. This can greatly reduce the effort and cost to do these tasks and make the automation of these tasks possible.

Reference

[Tsai 1999] W. T. Tsai, W. Shao, Y. Tu, and E. Ebner, "Testing Extensible Design Patterns in Object-Oriented Frameworks Through Hierarchical Scenario Templates" Proc. of IEEE COMPSAC 1999.

[Tsai 1999] W. T. Tsai, V. Agarwal, R. Paul, and B. Huang, "Augmenting Sequence Constraints in Z and its Application to Testing", Proc. of IEEE Assets, 1999,

[Vishnuvajjala 96] R. V. Vishnuvajjala, W.T. Tsai, "Specifying Timing Constraints in Real-Time Object-Oriented Systems", Proc. of High Assurance Systems Engineering (HASE) Workshop , Oct. 1996.

[Kirani 94a] S. H. Kirani and W. T. Tsai, "Method Sequence Specification and Verification of Classes", Journal of Object-Oriented Programming, Oct. 1994, pp. 28-38.

[Tsai 2002a] W. T. Tsai, L. Yu, R. Paul, T. Liu, and A. Saimi, "Developing Adaptive Test Frameworks for Testing State-based Embedded Systems", Proc. of IDPT, 2002.

[Tsai 2002b] W. T. Tsai, Y. Na, R. Paul, and F. Lu, "Adaptive Scenario-Based Object-Oriented Test frameworks for Testing Embedded Systems", to appear in Proc. of IEEE COMPSAC, 2002.

1530-2059/02 $17.00 © 2002 **IEEE**