

A Design of tool for Generating Web Service Stub for WS-BPEL

Nuttapong Wachiralohaphan¹ and Taratip Suwannasart²⁺

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand

Abstract. Service-Oriented Architecture (SOA) is a software architecture which is now widely applied for designing system architecture because it is an approach for designing independent services with well-defined invocable interfaces. As these results, web services have been widely developed. The trends in developing existing web services for achieving coordinated functions emerged in order to respond to the business needs. WS-BPEL has become popular because it is designed based on SOA and can be used to describe behavioral of business process on interaction between its partner services. In the software integration testing process, the problem usually found is that software tester cannot verify the request coverage of partner web services and thus it is necessary to operate web service stub to be substituted for actual web service. This paper presents a design of tool for generating web service stub for WS-BPEL. The tool can generate stub for web services, which are defined in business process, and also provides report to software tester to verify web service request coverage.

Keywords: Web Service, WS-BPEL, WSDL, Stub.

1. Introduction

Service Oriented-Architecture (SOA) [1] is an architectural software design approach that is widely used to design a software system nowadays because it consists of loosely coupled autonomous services. SOA allows developers to create services without specific programming languages. As these results, web services have been widely developed. Especially, in development of existing web services for achieving coordinated functions emerged in order to respond to the business needs. Business Process Description Language for Web Service (WS-BEPL) [2] has become popular because it is used for describing a business process which interacts with external web services.

Generally, in the software integration testing process. Software testers often face the problem with external services because it is in progress of development or unavailable, and software tester cannot verify the request coverage of partner web services and thus it is necessary to operate web service stubs to be substituted for actual web services in the software integration testing process.

The research [3] has proposed basis path test suite and testing process for WS-BPEL. The tool in this research also provides a feature for generating web service stub for WS-BPEL but this tool supports only basic data types excluding complex data type, and the web service stub that is generated by the tool can be defined response data only before installation of web service stubs.

In this paper, we aim to propose a design of tool for generating web service stub for WS-BPEL. Web service stubs are generated from WS-BPEL, WSDL, and XSD files. Web service stubs can randomly response data based on the schema of response data and it also allows software tester to configure response data manually in many types of responding data with no need to reinstall web service stubs. Moreover, the tool also provides reports, which help software testers check web service request coverage and web service request histories.

⁺ Corresponding author. Tel.: +66 2 2186956 ; fax: +66 2 2186955.
E-mail address: Nuttapong Wachiralohaphan (Nuttapong.Wa@student.chula.ac.th)

The rest of the paper is organized as follows. Section 2 introduces the background of WS-BPEL, WSDL, XSD, and Stub. Section 3 discusses about existing researches that related to an approach of generating web service stub of WS-BPEL and techniques for generating XML document based on XML Schema. Section 4 presents proposed design of our tool and Section 5 discusses conclusion and future work.

2. Background

2.1. WS-BPEL

WS-BPEL (Business Process Execution Language for Web Service) [2], which proposed by OASIS, is an XML-based language for describing behaviour of business process on interaction between its partner service. WS-BPEL process can be defined in two types of process. First, abstract process is used to describe some or all of the use cases that is not intended to be executed. And secondly, executable process is used to describe a use case of business process including interaction with all partner services.

WS-BPEL uses activity for performing process logic and divides activities into 2 classes. First, basic activities are used to describe elemental step of process behaviour such as invoking web service operations - invoke, providing web service operations – receive and reply, and updating variable and partner link - assign. And secondly, structured activities are used to describe control-flow logic of business process such as sequential processing – sequence, condition behaviour – if, repetitive execution – while, and parallel and control dependencies processing – flow.

WS-BPEL utilizes several XML specification documents such as WSD, XML Schema, XPath, to define the business process.

2.2. WSDL

WSDL (Web Service Description Language) [4] is an XML-based language which is used for describing the web service. WSDL can be defined in two types. First, abstract type is used to describe message, data type definition of message, and operation. And secondly, concrete type is used to describe message format and network protocol for operations, and endpoint of services.

WSDL document is also used for describing the interaction between business process and its partner web services by adding some special xml elements, which defined by OASIS standard, to define partner link type and role.

2.3. XSD

XSD (XML Schema Definition Language) [5] [6] is an XML-based language that is used to define the schema of XML document including element and attribute that can be defined in the xml document, data types of attribute and element, restrictions which are used to define acceptable value of attribute and element, and child elements. Additionally, XSD can be used to validate the correctness of XML document against the XML schema. XSD is also used for describing the schema of XML document in another document such as WSDL, WS-BPEL.

2.4. Stub

Stub [7] is a technique that is used in software integration testing process in top-down approach. Stub simulates behaviors of software component or modules and is used to substitute actual web service not only in the testing process but also in the development process in case of the actual web service is unavailable or under development. Most of stub usually record the request data into the database and used it for rendering report to user.

3. Related Work

The research on software testing of the business process has become a popular topic in the recent years. T. Lertphumpanya et al. [3] has proposed basis path test suite and testing process for WS-BPEL and has developed tool by providing a feature for generating web service stub for WS-BPEL for testing the basis path. For the research on generating XML document from XML schema, D. Petrova-antonova et al. [8] has proposed an approach for generating test data based on XML schema and create a tool to prove the concept.

This research divided the process of generating test data into 2 groups. First. The process of analyzing structure of XML schema and secondly, the process of generating test data from XML schema. A. Bertolino et al. [9] has proposed XML-based Partition Testing (XPT) approach for automatically generating XML instances from XML Schema. This research described mapping of the XML Schema to the category partition concepts, which leads to the generation of an unmanageable number of XML instances.

According to the researches above, we found some restrictions of the tool for generating web service stub for WS-BPEL that is the response data of web service stubs can be defined only before installation of web service stubs. If users want to test the business process with a new test case, they have to generate and install the web service stub once more. Our proposed approach comes up with an idea in creating a tool that can generate web service stubs for WS-BPEL which can manage the response data of web service stub and do not require reinstallation of web service when the response data is changed.

4. Proposed Approach

We propose a design of tool for generating web service stub for WS-BEPL. The tool contains 3 main parts as illustrated in Fig 1. First, generating web service stub part is the process of generating web service stub from imported files. Second, web service stub part is responsible for providing function for managing response data of web service stub and responding data to service requester. And lastly, reporting part is responsible for generating reports to software testers.

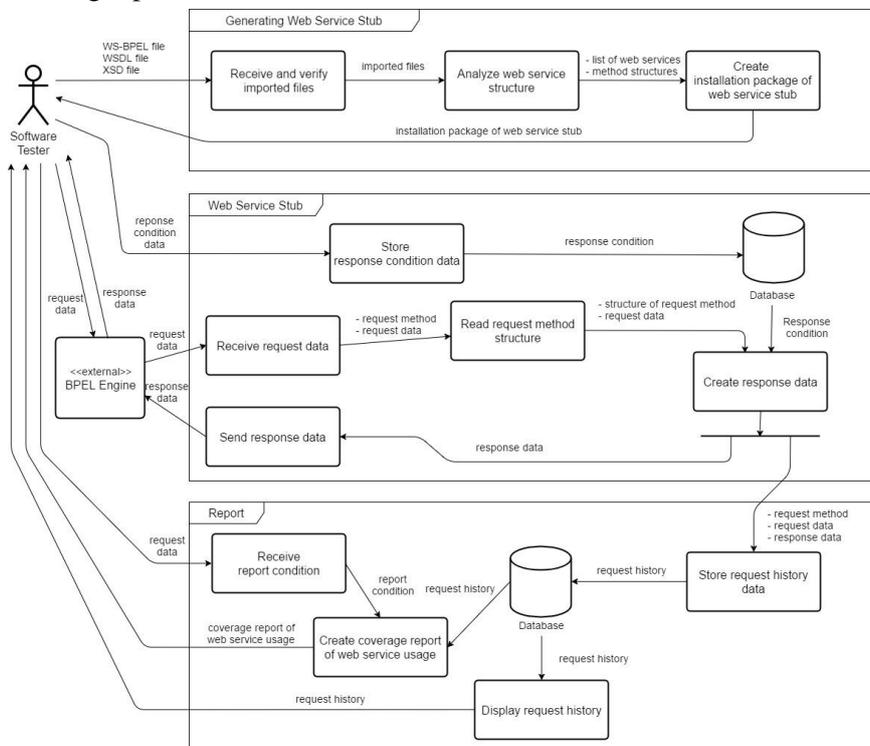


Fig. 1: A design of tool for generating web service stub for WS-BPEL.

4.1. Generating web service stub

This part is responsible for generating web service stub for WS-BPEL from imported files that contains two modules as follows.

4.1.1. Analysing imported data module

This module analyses imported files from a tester, and consists of 3 types of files including WS-BPEL, WSDL, and XSD, for validating format and finding relation of data from imported files. If the validation process is passed, it will process 3 steps below.

- Find basic activity nodes ('invoke' and 'receive'), which are responsible for calling and receiving data between its partner web service, in business process that is defined in WS-BPEL file.
- Read the structure of web services, which are used by business process, in WSDL files.
- Read the schema of web service response data in schema definition in WS-BPEL file or XSD file.

For example, a business process of vehicle information from WS-BPEL file is illustrated in Fig 2. This business process describes a process for retrieving vehicle information including price that is calculated from its pricing rule. This business process contains 3 partner web services. First, vehicle information web service is used for retrieving vehicle information and vehicle pricing rule. And secondly and lastly, escalator and rounder web services are used for retrieving vehicle price and one of these two web services will be chosen for retrieving price based on pricing rule of vehicle. This module will process the steps above and prepare the data from Table 1 for generating web service stub in the next module.

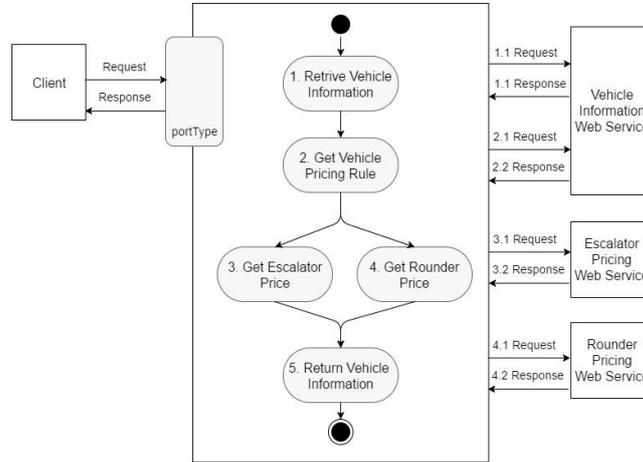


Fig. 2: An example of a business process.

Table 1: Web service information of an example business process.

Web Service	Operation	Request Data Type	Response Data Type
Vehicle Web Service	GetVehicleInformation	VehicleId	VehicleInfo
Vehicle Web Service	GetPricingRule	VehicleId	PricingRuleClass
Escalator Pricing Web Service	GetPrice	VehiclePrice	Decimal
Rounder Pricing Web Service	GetPrice	VehiclePrice	Decimal

4.1.2. Generating web service stub module

This module uses the data from previous module to generate the web service stub, which is developed to support both synchronous and asynchronous types of web services, and creates the installation package that contains 2 parts below.

- Web service stubs which consist of all partner web services in business process.
- Web application for managing response data of web service, monitoring web service request-response data and inquiring reports.

4.2. Web service stub

This part is responsible for providing response data management for web service stub and responding data to business process which contains two modules.

4.2.1. Managing condition of response data module

This module is designed to allow testers to configure the response data of web service stub on the web application which helps user reduce the installation time of web service stub when the testers wants to change the response data of web service stub. This module provides types for responding data of web service stub as follow.

- Randomly respond data based on XML schema.
- Randomly respond data from user-defined data set.
- Sequentially respond data from user-defined data set.
- Randomly respond data from user-defined weighted data set.

This module utilizes the XML schema, which is defined in schema definition of both XSD file and WSDL file, and allow testers to input the response data based on the XML schema of web service operation.

4.2.2. Responding data module

This module is in charge of responding data when there is incoming request from business process to web service stub. It is process the steps below.

- Receive web service request from business process.
- Read the configured data of web service operation which is defined on the web application.
- Read the schema of response data.
- Create response data based on the type of responding data and the schema of response data.
- Store the request-response data into the database for generating reports.
- Send the response data to the business process.

4.3. Reporting

The objective of this module is for providing report to the software testers for tracking the request of web service stub and checking the request coverage status of web service stub. The tool provides 2 types of report below.

- Web service stub request history report: used for tracking the request data such as web service name, operation name, request time, request data, response data and processing time.
- Web service stub request coverage report: used for verifying the coverage status of web service request to web service stubs

5. Conclusion and Future Work

This paper proposed a design of tool for generating web service stub for WS-BPEL. The tool can generate web service stubs of all partner web services in the business process by analysing business process in WS-BPEL file, and read the structure of web service in WSDL files and read schema of web service response data in XSD files. The tool provides function for managing response data of web service stub in many types. Additionally, the tool also provides report to software testers for checking the request history and request coverage.

For future work, we will develop a tool based on our proposed design and test it with business processes which are selected from industry business processes.

6. References

- [1] "Reference Model for Service Oriented Architecture 1.0." [Online]. Available: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html>. [Accessed: 1-May-2017].
- [2] "Web Services Business Process Execution Language Version 2.0." [Online]. Available: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>. [Accessed: 1-May-2017].
- [3] T. Lertphumpanya and T. Senivongse, "Basis path test suite and testing process for WS-BPEL," WSEAS Trans. Comput., vol. 7, no. 5, pp. 83–96, 2008.
- [4] "Web Services Description Language (WSDL) 1.1." [Online]. Available: <https://www.w3.org/TR/2001/NOTE-wsdl-20010315>. [Accessed: 1-May-2017].
- [5] "W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures." [Online]. Available: <https://www.w3.org/TR/2012/REC-xmlschema11-1-20120405>. [Accessed: 1-May-2017].
- [6] "W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes." [Online]. Available: <https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405>. [Accessed: 1-May-2017].
- [7] P. C. Jorgensen, Software Testing: A Craftsman's Approach 4th ed. pp. 183-194, 2004.
- [8] D. Petrova-antonova, K. Kuncheva, and S. Ilieva, "Automatic Generation of Test Data for XML Schema-based Testing of Web Services," in Software Technologies (ICSOFTE), 2015 10th International Joint Conference on, 2015.
- [9] A. Bertolino, J. Gao, E. Marchetti, and A. Polini, "Automatic Test Data Generation for XML Schema-based Partition Testing," in Second International Workshop on Automation of Software Test (AST '07), 2007, pp. 4–4.