Chapter 1

Complex Network Theory Based Web Services Composition Benchmark Toolkit

Seog-Chan Oh
General Motors R&D Center, USA

Dongwon Lee
The Pennsylvania State University, USA

ABSTRACT

In recent years, while many research proposals have been made toward novel algorithmic solutions of a myriad of web services composition problems, their validation has been less than satisfactory. One of the reasons for this problem is the lack of real benchmark web services data with which researchers can test and verify their proposals. In this chapter, to remedy this challenge, we present a novel benchmark toolkit, WSBen, which is capable of generating synthetic web services data with diverse scenarios and configurations using complex network theory. Web services researchers therefore can evaluate their web services discovery and composition algorithms in a more systematic fashion. The development of WSBen is inspired by our preliminary study on real-world web services crawled from the Web. The proposed WSBen can: (1) generate a collection of synthetic web services files in the WSDL format conforming to diverse complex network characteristics; (2) generate queries and ground truth sets for testing discovery and composition algorithms; (3) prepare auxiliary files to help further statistical analysis; (4) convert WSDL test sets to the formats that conventional AI planners can read; and (5) provide a graphical interface to control all these functions. To illustrate the application of the WSBen, in addition, we present case studies selected from three domains: (1) web services composition; (2) AI planning; and (3) the laws of networks in Physics community. The WSBen toolkit is available at: http://pike.psu.edu/sw/wsben/. This chapter is an invited extension of authors’ previous publication (Oh & Lee, 2009).

DOI: 10.4018/978-1-61350-104-7.ch001
INTRODUCTION

A Web Service is a set of related functionalities that can be loosely coupled with other (web) services on the Web. As long as a web application exposes its functionalities using standard-based web service API such as the XML-based WSDL, it can be used as a web service by other web applications. This openness, platform-independence, and reusability of web services therefore pose tremendous opportunities for web commerce and businesses, enabling web services framework the de facto standard for “accessing” applications and “sharing” data on the Web. Together with the recent surge of interest on the Cloud Computing, the popularity of web services in industries will become even stronger in the near future since virtually all applications in the Cloud Computing can be treated as web services applications.

As a growing number of web services are available on the Web and in organizations, finding and composing the right set of web services for certain tasks become an increasingly important problem. As a result, in recent years, a plethora of research work and products on web services discovery and composition problems have appeared. As of November 2010, according to the estimation of Google Scholar, there are about 1,870,000 scholarly articles mentioning “Web Services Composition”. In addition, the web service research community has hosted several open competition programs to solicit algorithms and software to discover pertinent web services and compose them to make value-added functionality as follows:


Despite all this attention, however, there have been very few test environments available for evaluating such algorithms and software. The lack of such a testing environment with flexible features hinders the development of new composition algorithms and validation of the proposed ones. Therefore, the need for a benchmark arises naturally to evaluate and compare algorithms and software for the web services discovery and composition problems. As desiderata for such a benchmark, it must have (a large number of) web services in the standard-based WSDL files and test queries that can represent diverse scenarios and situations that emphasize different aspects of various web services application domains. Often, however, test environments used in research and evaluation have skewed test cases that do not necessarily capture real scenarios. Consider the following example.

Example 1 (Motivating) Let us use the following notations: A web service \( w \in W \), specified in a WSDL file, can be viewed as a collection of operations, each of which in turn consists of input and output parameters. When an operation \( o \) has input parameters \( o^p = \{p_1, \ldots, p_n\} \) and output parameters \( o^o = \{q_1, \ldots, q_n\} \), we denote the operation by \( o^p(o^i, o^o) \). Furthermore, each parameter is viewed as a pair of (name, type). We denote the name and type of a parameter \( p \) by \( p.name \) and \( p.type \), respectively. For establishing our motivation, we first downloaded 1,544 raw WSDL files that Fan et al. (Fan & Kambhampati, 2005) gathered from real-world web services registries such as XMethods or BindingPoint. We refer to the data set as PUB06. For the purpose of preprocessing PUB06, first, we conducted WSDL validation according to WSDL standard, where 874 invalid WSDL files are removed and 670 files are left out. Second, we removed 101 duplicated WSDL files at operation level, yielding 569 valid WSDL files. Finally, we conducted type flattening and data cleaning processes subsequently. The type flattening process is to extract atomic types from user-defined complex types using type hierarchy of