A Practical Approach Towards Automatic Testing of Web Services Interoperability

Ivano Alessandro Elia, CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

Nuno Laranjeiro, CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

Marco Vieira, CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

ABSTRACT

Web Services are a technology designed to support the invocation of remote elements by client applications, with the goal of providing interoperable application-to-application interaction while assuring vendor and platform independence. In business-critical environments, the occurrence of interoperability issues can have disastrous consequences, including direct financial costs, reputation, and client fidelity losses. Despite this, experience shows that services interoperability is still quite difficult to achieve. The goal of this paper is to propose a practical testing process to understand the real level of interoperability provided by web services platforms. An extensible tool, that implements the proposed approach, has been used to run a large campaign during which we have tested the interoperability of a large number of web services, comprising both home-implemented and publicly available services, deployed on top of several web service platforms and against 11 client-side web service platforms. Numerous issues have been disclosed, showing the effectiveness of the proposed approach and the importance of having an automatic tool for testing web services for interoperability.

Keywords: Interoperability, Testing, Testing Tools, Web Service

1. INTRODUCTION

Web services are being increasingly used to integrate enterprise systems, linking complex applications in large-scale organizations and supporting critical business-to-business interactions, where the typically heterogeneous environments frequently demand a strict definition of service interfaces and interaction patterns.

In such heterogeneous scenarios, systems designers frequently opt for using web services technology, as it was created with the key goal of supporting the inter-operation between different systems, including systems built using different programming languages (Erl, 2005).

In a typical services environment a service provider announces its interface using a WSDL document that describes, in a programming language agnostic way, the operations available...
able to the clients. In turn, clients make use of the interface description to understand how the service operations can be invoked (Curbera et al., 2002). Nowadays, developers do not need to implement all supporting code for generating services or invoking operations, as most web service frameworks (platforms for creating, deploying, and invoking web services) provide automatic code generation facilities and runtime support for invoking service operations (Hewitt, 2009). The problem is that, although web services are supported by machine independent protocols and have been designed with interoperability in mind, practice shows that it still is quite difficult to link distinct applications. This is especially true in complex and more heterogeneous scenarios, such as the ones frequently found in enterprise environments.

The interoperability of web services is an issue in which the Web Services Interoperability Organization (WS-I) (Web Services Interoperability Organization (WS-I), 2002) has been working for several years. However, practice shows that developers still struggle to deploy services that can fully inter-operate with client applications. In fact, developers many times create and deploy their web services expecting the underlying framework to provide full interoperability, which is not always the case. Our tests, presented later in the paper, confirm this scenario, showing that the inter-operation between different frameworks is not yet fully achieved and highlighting many cases where inter-operation is not possible.

A key problem is that, although efforts have been undertaken towards creating tools to test services (Eviware, 2011, WS-I, CrossCheck Networks), they appear to be quite limited, when it comes to interoperability testing. Among all tools, those provided by the WS-I Testing Tools Working Group (“WS-I,”) are definitely a step forward, but experience suggests that even WS-I compliant services may show interoperability issues. Thus, there is a clear need for a specialized approach for interoperability testing that helps developers conducting broader interoperability assessments, from a practical perspective.

We presented a tool for interoperability testing of web services in (Elia, Laranjeiro, & Vieira, 2014b). In (Elia, Laranjeiro, & Vieira, 2014c) and (Elia, Laranjeiro, & Vieira, 2014a) we have conducted experiments regarding part of the full inter-operation process (which in the end involves communication). In this paper we greatly extend the previous work, by having refined the approach and formalized key aspects, extended the approach to now include the exchange of messages between client and server, and by augmenting the tests and tool implementation to cover the approach extension and thus covering the whole inter-operation process.

We propose, in this paper, a practical approach to support the evaluation of all the steps of the inter-operation process in web services environments. We also present ITWS (Interoperability Testing for Web Services), an open-source web services testing tool available at (Elia, Laranjeiro, & Vieira) that supports the execution of the proposed approach. Developers can use ITWS to evaluate the level of interoperability attained by their services (running in the context of a given framework, i.e., a platform that provides the deployment and runtime support for web services) with regard to one or more client-side frameworks (that offer client-side support for invoking the web services operations). ITWS currently supports testing a set of well-known web service frameworks, but the tool was designed to be extensible and adding more frameworks requires only a simple configuration effort.

The ITWS tool was used to run a massive test campaign involving about 8 thousand web services, comprising over seven thousand home developed services deployed on top of three major WS frameworks and over one thousand web services publicly available on the Internet and also deployed on top of multiple frameworks leading to a total of 174547 interoperability tests.

The outline of this paper is as follows. The next section presents the web services inter-operation problem and discusses existing techniques and tools for interoperability testing. Section 3 describes our approach for testing
Satisfying End User Constraints in Service Composition by Applying Stochastic Search Methods
www.igi-global.com/chapter/satisfying-end-user-constraints-service/66962?camid=4v1a