ELEKTROTEHNIŠKI VESTNIK 79(1-2): 61-67, 2012
ENGLISH EDITION

Assessing the usefulness of the WS-I tools for interoperability testing

Tomaž Korelič, Marjan Heričko

University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia
E-mail: tomaz.korelic@uni-mb.si

Abstract. Interoperability is a key feature when it comes to the successful communication of components implemented in different technologies. This paper presents the WS-I Basic Profile 1.2 (BP1.2) and the corresponding testing tools to test interoperability of web services. We conducted an experiment in which we implemented several web services in different frameworks and evaluated their level of interoperability. Using the WS-I testing tools, we validated the web service artifacts for conformance with BP1.2. We will present our experiences using the WS-I testing tools and review their usefulness in detailing the problems that occurred. The results indicate that no interoperability issues were found while using modern web service frameworks.

Keywords: interoperability, web services, WS-I, Basic Profile 1.2, testing

1 INTRODUCTION

Interoperability refers to the ability of software and hardware on multiple machines from multiple vendors to communicate with each other without significant changes on either side [1]. Even though the established standards for web service messaging (SOAP), description (WSDL) and registry (UDDI), tried to solve interoperability issues, custom implementations of vendors have not solved them completely [1].

1.1 The Web Service Interoperability Organization (WS-I)

The Web Service Interoperability Organization (WS-I) extracted the best practices and issues of web service interoperability from several standards, platforms, operating systems and programming languages and merged them into several profiles. These included implementation guidelines for web services to assure interoperability. The WS-I prepared the Basic Profile 1.1, 1.2 and 2.0, the Attachments Profile 1.0, the Simple SOAP Binding Profile 1.0, the Reliable Secure Profile 1.0, the Kerberos Token Profile 1.0, the REL Token Profile 1.0, the SAML Token Profile 1.0 and the Basic Security Profile 1.1 [2].

The profiles include conformance requirements and the corresponding test cases validate the conformance of test artifacts included in SOAP messages and the WSDL document. However, the execution of interoperability testing results in a conformance report with the profile does not guarantee 100% interoperability with other platforms [3].

The results of the WS-I's work were acknowledged by many, especially during the preparation of new standards like SOAP 1.2 [1]. In November 2010, WS-I completed their work on interoperability standards. Their assets, operations and mission transitioned to OASIS [4].

This paper introduces the WS-I Basic Profile 1.2 and the testing tools for determining the level of conformance. Many web service vendors claim conformance with the BP. During our experiment, we tried to confirm their statements by implementing several web services and invoking them among them. During this execution, we checked for conformance of the created artifacts with the BP1.2 using the WS-I testing tools and confirmed their interoperability. The goal of this research was to obtain and report the experiences of using the WS-I testing tools and the BP1.2. Based on the provided results, we determined the usefulness of these tools, using them on modern web service frameworks.

1.2 The WS- Standards*

Interoperability issues are also addressed by the WS-* standards. HTTP, HTTPS and SMTP manage the transport of messages, which are defined using different XML standards (including XSL, Xpath and others), SOAP, *Attachments for SOAP* and WS-Addressing for reliable service addressing. Services are described using WSDL, registered using UDDI, WS-Inspection and WS-MetadataExchange. WS-ReliableMessaging handles the reliable delivery of messages, WS-Security safe communication, WS-Transaction guarantees the successful transaction execution and WS-BPEL deals with service composition execution [5]. Although these

Received May 6, 2012
Approved May 15, 2012

standards are widely adopted in practice, interoperability issues still remain because of different custom implementations. The WS-I profiles adopt these core web service standards, with the exception of WS-Transaction and WS-BPEL.

2 Related Work

Kumar et al. [1] presented some key interoperability issues in practice and analyzed how Basic Profile 1.0 (BP) addresses them. They evaluated the limitations of the profile using the most popular web service frameworks at the time. One limitation was revealed with regard to the Basic Profile not testing the interoperability of datatypes. Additional problems were identified using datatypes in XML schemas in SOAP messages, SOAPAction headers and the extensibility of particular elements in WSDL1.1. The findings refer to BP1.0 from the year 2004, but our research focuses on BP1.2 from 2010. They used the same tools for testing interoperability of the Apache AXIS1.0 framework. The results indicated that AXIS1.0 was not conformant with BP1.0. Our experiment also included AXIS2 1.6.1.

Bertolino et.al. [6] proposed a new framework for the interoperability testing of web services based on protocol state machine diagrams. The framework evaluates the sequence of invocations between two different web services and tries to discover any possible interoperability issues. However, the authors did not use the WS-I testing tools, like we did. The proposed solution successfully addresses interoperability issues using a new approach.

Simon et al. [7] prepared a test case suite that evaluates interoperability of SOA products based on WS-* standards. The automatic test cases are reusable and the test framework is flexible enough to include new web services and integrate them with existing ones. The results were evaluated for several web service frameworks.

Kuppuraju et al. [8] proposed a new methodology to verify interoperability of several products. During their case study, they implemented and evaluated the methodology on multiple web services. Using the WS-I testing tools they insured that no interoperability issues were present before the evaluation. They do not report their personal experiences using the profile or the testing tools.

Papastergiou et al [9] introduced an e-invoicing system based on web services that achieved a high level of interoperability. They passed the majority of conformance tests for several profiles using the WS-I testing tools. It was also emphasized that WS-I profiles do not cover all interoperability issues, because they primarily cover the exchanged messages. Additional interoperability problems were reported regarding web service security and digital signatures in SOAP messages. Their web services were identified with a medium to high level of interoperability, based on conformance testing with BP1.0.

We identified two kinds of studies that test the interoperability of web services. The first group consisted of studies that use the WS-I testing tools to test interoperability [1] [8] [9]. Some authors reported several flaws with the profiles and conformance problems with the tested frameworks. Our study differs from theirs in the version of the profiles and testing tools. The second group of studies proposed new testing methodologies, without the use of the WS-I's tools [6] [7]. Based on the information found in those studies, we prepared the test cases for our web services. Our study also addresses some usability issues when using the WS-I testing tools and the BP1.2.

3 The WS-I Profiles

The WS-I testing tools check the conformance of WSDL documents, SOAP messages and XML schemas with the profiles. Our research focuses on the Basic Profile 1.2 [3], although other profiles exist which validate the conformance of other targets like WS-Reliable Messaging and WS-Security. WS-I also introduced the Basic Profile 2.0, which boasts the same requirements, but validates conformance with Soap1.2. BP1.2 uses Soap1.1. Since we enabled the use of Soap1.1 in our test web services, we chose BP 1.2 [3].

3.1 Profile structure

Each profile is constructed in a similar way. Its requirements are presented in the XML format. A separate TAD (Test Assertion Document) document includes test cases which check conformance with the requirements using Xpath [10][11]. BP1.2 and BP2.0 have the TAD document already bundled in. The profile explicitly states that it does not guarantee interoperability, but rather addresses common interoperability issues.

The first chapter of the Basic Profile 1.2 addresses its common properties and the second describes what and how it is being tested. The concluding chapters include specific conformance requirements with the corresponding test artifacts and test cases. The following are a few examples of some BP1.2 requirements:

R9980 *- An ENVELOPE MUST conform to the structure specified in SOAP 1.1 Section 4, "SOAP Envelope" (subject to amendment by the Profile).*
R1141 *- When HTTP is used as the transport, a MESSAGE MUST be sent using either HTTP/1.1 or HTTP/1.0.*
R1040 *- If an endpoint requires use of WS-Addressing by use of a wsam:Addressing policy assertion, an ENVELOPE sent by a SENDER MUST carry all required WS-Addressing SOAP headers.*

3.2 Test cases in BP1.2

BP1.2 is a set of recommendations and guidelines covering the core web service standards including SOAP 1.1 over HTTP1.1, WS-Addressing 1.0, message serialization, SOAP envelope, SOAP failures and use of URIs.

The test cases check the conformance of SOAP messages, WSDL documents and XSD schemas for the correct use of *HTTP/1.1*, message serialization, SOAP envelope, SOAP faults, use of SOAP in HTTP, WS-Addressing. The WSDL test cases validate such elements as *portType*, *Messages*, *types*, *SOAP Binding* and the WS-Addressing metadata.

The correct use of datatypes is tested only for *Array* in *anyType*. Additional conformance tests for WSLD documents check the conformance of portTypes, Binding, XML Schemas and WS-Addressing. Chapters 5 and 6 of the profile include requirements for UDDI and the use of HTTPS, but conduct no test cases to validate them. In fact only 66% of requirements include the corresponding test cases that automatically test conformance [12].

3.3 WS-I testing tools

WS-I implemented two tools (Monitor and Analyzer) that are meant to be used together for testing conformance. The latest version 1.2 (and 2.0 for BP2.0) of the testing tools differs significantly from the older ones in setup and usage, but the functionalities remain the same. They require the Apache web server, Python and Perl. The older versions were executed from the command line.

The Monitor is a component that acts as a *man in the middle* and intercepts the exchanged SOAP messages between the web service and the client through the HTTP protocol. After the interception, it creates a Test Log File using the SOAP messages, the WSDL document and the XML schemas. The Analyzer receives that Test Log File, WSDL and XML schemas as an input and analyzes all test artifacts for conformance with the provided profile. The conformance report is finally generated [13]. Figure 1 details the execution of the WS-I Testing Tools.

The Conformance Test Report includes details about the executed test cases on specific test artifacts and their results. The possible conformance results are *Successful*, *Failed*, *Not relevant*, *Missing input* and *Warning*. If the Test Report does not include any test cases with the result *Failed*, then we assume that the web service conforms with the profile [14].

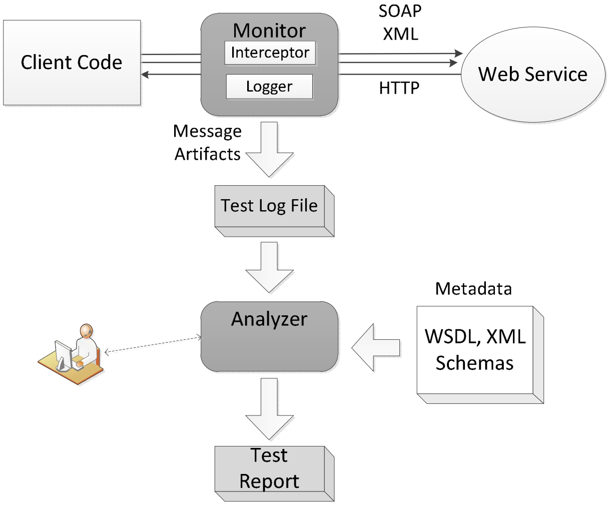

Figure 1: WS-I Testing Tools schema

4 Experiment

With the experiment we would like to confirm whether the web services that are implemented using modern frameworks conform with BP1.2. We chose four modern frameworks and implemented test web services in each one. We prepared four test case scenarios for invoking the test web services. During this time, we validated conformance with BP1.2 using the WS-I Testing Tools. We invoked them using SoapUI in the first step and between each other in the second. This way we confirmed that our web services were interoperable, causing no problems while invoking them across platforms. During the experiment we gained some experience with using the testing tools, which was the main goal of our research.

All major vendors claim conformance with the WS-I profiles. The JAX-WS project develops and evolves the code base for the reference implementation of the Java API for XML Web Services and is the core of the Metro Project inside the Glassfish community. Officially JAX-WS supports Basic Profile 1.2 and 2.0 [15]. Project Metro includes the WSIT project (Web Services Interoperability Technologies), which includes the implementation of WS-Trust, WS-SecureConversation, WS-SecurityPolicy, WS-ReliableMessaging, WS-AtomicTransactions/Coordination, WS-MetadataExchange and SOAP over TCP.

Project Metro was tested for interoperability between the platform Java 6 and Windows Communication Foundation (WCF) on .NET 3.0 and 3.5 [16].

Microsoft states that WCF WS-*, as part of .NET 4, is conformant with all WS-I profiles and is interoperable with all major web service vendors (Oracle, IBM, SAP, Project Metro) [17]. All interoperability settings are accessible in the *web.config* file inside a WCF project in Visual Studio 2010.

The CXF framework implements the JAX-WS API. On their official website, it states that it is conformant with Basic Profile 1.1, and that it supports WS-Addressing, Soap 1.1 and Soap 1.2 [18].

The official website of the AXIS2 framework does not contain any information about being conformant with any WS-I profile [19]. It officially supports the standards SOAP1.1 and SOAP1.2, WSDL 1.1 (SOAP and http binding), WS-Addressing, WS-policy and transport over http, smtp, jms and tcp. Based on these facts, we can assume that AXIS2 conforms with the Basic Profile 1.2.

Most web service vendors claim that their framework is at least conformant with the WS-I Basic Profile 1.0 or BP1.2. Using our experiment, we would like to confirm this fact, and gain more experience using the WS-I Testing Tools.

4.1 Test web services

We chose four web service frameworks to implement the test web services: WCF 4, Jax-WS 2.2.5, CXF 2.4. and Axis2 1.6.1. The implemented operations covered the major requirements included in the BP1.2 together with WS-Addressing. Table 1 shows an overview of all the chosen web service frameworks, including the application server, IDE, programming language and configuration type. All the web services were implemented using the contract-last approach, where the WSDL document is generated automatically.

Table 1: Overview of the test frameworks

WS-API and vendor name	Application Server	IDE	Programming language	Configuration
WCF 4 *Microsoft*	asp.net development Server 4.0.30319.237	Visual studio 2010	C#	Custom XML
Jax-ws 2.2.5 *Sun*	Glassfish 3.1	Netbeans 7.0	Java	WS-Policy
CXF 2.4.3 *RedHat*	JBoss AS 7.0.2	Eclipse indigo	Java	Custom XML, WS-Policy
Apache Axis 2 (1.6.1) *Apache*	Tomcat 6.0.14	Eclipse indigo	Java	Custom XML

4.2 Test cases

All test cases were prepared based on Simon et al. [7]. They tested the interoperability of web services based on the WS-* standards using their own test suite, without the use of the WS-I profiles. Regardless of this fact, they provided solid guidelines for our research. Their execution covers the majority of the generated artifacts in SOAP messages that are evaluated during conformance testing. If we stick to their guidelines for preparing test cases, all frameworks will generate the most appropriate artifacts for the conformance testing of SOAP messages and WSDL documents with the WS-I Basic Profile 1.2. The conformance test cases are very technical and validate the correct structure of the test artifacts which were generated automatically. We did not prepare one test case for each requirement, but rather covered the profile’s multiple topics in separate test cases. Our research focused on SOAP 1.1 over HTTP, WS-Addressing, fault handling and the use of both datatypes that the profile validates. All web services were enabled to use WS-Adressing 2005-08.

The Basic profile 1.2 does not validate the correct use of all complex datatypes. Despite this fact, we included the the datatype *Date* in one scenario to obtain information about the correct conversion between the platforms. Because BP1.2 validates the datatype *List* [3], we included it in one operation and a corresponding test case was prepared using *ArrayList<String>*.

In summary, we prepared web services in four different frameworks on two platforms. The following test cases were executed from different clients to test conformance with BP1.2:

- Invoke the operation *float* Divide(54.3, 3.0) using WS-Addressing 2005-08.
- Invoke the operation *int* DivideInteger(54, 0) z WS-Addressing 2005-08. Expected result: exception (*Divide with 0*).
- Invoke the operation *Date* GetCurrentDate() z WS-Addressing 2005-08. Expected result: returns the current date.
- Invoke the operation *ArrayList<String>* GetEventList(Date) z WS-Addressing 2005-08. Expected result: returns an array of three events on a specific date (5.11.2011).

4.3 Used tools

During out experiment, we used the following tools: WS-I Monitor 1.2, WS-I Analyzer 1.2, TCPMon 1.0, WSIHero 1.0 and SoapUI 4.0. The web services were implemented in Visual Studio 2010, Netbeans 7.0 and Eclipse Indigo.

4.4 Execution

The first stage of our experiment consisted of invoking the test web services from SoapUI 4.0 [20] using the Monitor tool as a proxy server on port 8181. SoapUI enables all communication to be forwarded to a proxy server. The test web services were deployed on separate application servers. After the execution of all test cases for one platform, the Monitor created a log file that was forwarded to the Analyzer. Based on that log file, as well as the WSDL document and the XSD schemas, the analyzer generated a conformance report. Figure 2 depicts the deployment diagram of the first stage of the experiment, including all application servers, the SoapUI tool and the proxy server (Monitor). During the second stage we invoked the web services between each other. We prepared the clients in all frameworks and invoked all other web services from them including itself (Figure 3).

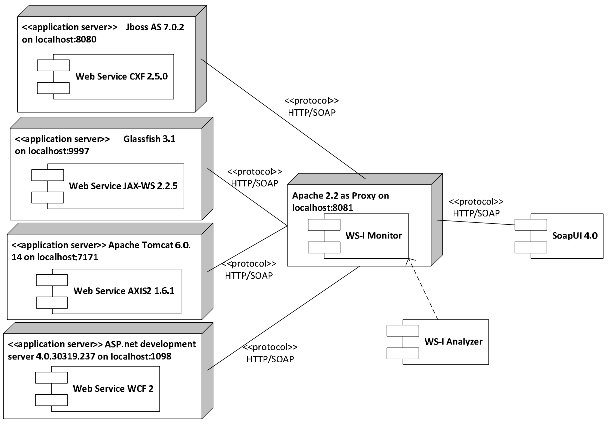
<<application server>> Jboss AS 7.0.2 on localhost:8080
Web Service CXF 2.5.0
<<application server>> Glassfish 3.1 on localhost:9997
Web Service JAX-WS 2.2.5
<<application server>> Apache Tomcat 6.0.14 on localhost:7171
Web Service AXIS2 1.6.1
<<application server>> ASP.net development server 4.0.30319.237 on localhost:1098
Web Service WCF 2
<<protocol>> HTTP/SOAP
<<protocol>> HTTP/SOAP
<<protocol>> HTTP/SOAP
<<protocol>> HTTP/SOAP
Apache 2.2 as Proxy on localhost:8081
WS-I Monitor
<<protocol>> HTTP/SOAP
SoapUI 4.0
WS-I Analyzer

Figure 2: Schema of the experiment - first stage

Because of technical difficulties in setting up the Monitor as a proxy server for the IDEs during the cross-platform invoking stage, we had to use an alternative. The tool Tcpmon 1.0 [21] replaced the Monitor during this stage, because it failed to intercept the SOAP messages. Tcpmon intercepted all http traffic, from which we manually extracted the SOAP messages and forwarded them into WsiHero [22]. This tool makes it easier to use the Analyzer using a graphical user interface. WsiHero uses the Analyzer in the background during conformance testing, but also includes some drawbacks. It accepts SOAP messages in the XML format, without the accompanying Http header, which is required in some conformance test cases. Those test cases failed during execution, precisely for this reason.

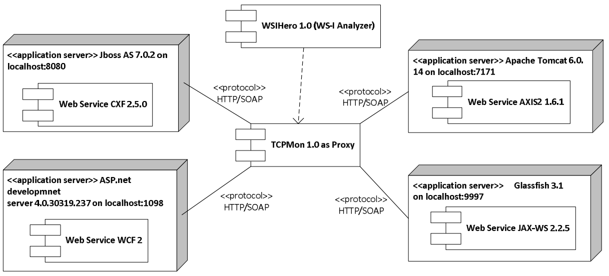
WSIHero 1.0 (WS-I Analyzer)
<<application server>> Jboss AS 7.0.2 on localhost:8080
Web Service CXF 2.5.0
<<protocol>> HTTP/SOAP
<<protocol>> HTTP/SOAP
<<application server>> Apache Tomcat 6.0.14 on localhost:7171
Web Service AXIS2 1.6.1
TCPMon 1.0 as Proxy
<<application server>> ASP.net developmnet server 4.0.30319.237 on localhost:1098
Web Service WCF 2
<<protocol>> HTTP/SOAP
<<protocol>> HTTP/SOAP
<<application server>> Glassfish 3.1 on localhost:9997
Web Service JAX-WS 2.2.5

Figure 3: Schema of the experiment - second stage

4.5 Results

The conformance test report is split into two parts. The first part consists of the executed test cases on all SOAP messages and the second on the WSDL documents and XSD schemas. The WSDL document is always the same in all test cases on each platform.

The conformance test cases were executed successfully on all WSDL documents. Consequently, all four WSDL documents were 100% conformant with Bthe asic Profile 1.2. Figure 4 presents conformance of SOAP messages from different clients. In the case of SoapUI as a client, the majority of all conformance test cases were executed successfully and none failed. In other clients, the test cases BP1901 or BP1905 returned a warning. They refer to an allowed, but not recommended use of a description element at an exception.

The JAX-WS service returned two additional warnings: BP1090a and BP1153a. BP1090a refers to the *wsa:Action* element in the SOAP message that included the error. The element obtained the value http://www.w3.org/2005/08/addressing/fault. BP1153a refers to the possible absence of the wsa:To element. After a manual check, we confirmed that the element was in fact present.

As presented in the graphs (Figure 4), the JAX-WS framework was, overall, the most conformant with BP1.2. It had the largest share of successful test cases. On the other hand, this service also had the most warnings. Since we designed our test cases based on the topics of the profile, and not on each requirement, some conformance test cases did not execute or were not relevant. Their conformance result was *missing input* coloured in dark red. Such a result occurs when the primary type of data was not specified as an input to the Analyzer or was not found in the tested artifact [14]. The AXIS2 service had the largest amount of such results.

The results from the first stage of the experiment presented a benchmark for the second stage, because all web services were conformant with BP1.2.

During the cross-platform invocations, some of the conformance test cases failed. BP1126a, BP1143c and BP1144 failed because of the absence of an Http header, which is a drawback of WsiHero (as presented in Chapter 4.4). For the same reason, BP1007, BP1019, BP1153a, BP1208 and BP1600 returned with *missing input*. More detailed results are presented in our technical report [23].

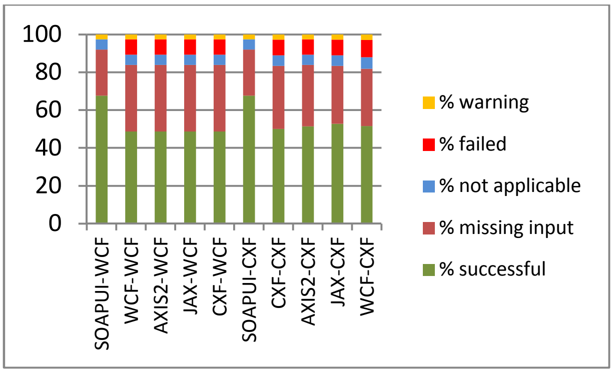

Figure 4: Conformance of SOAP messages for WCF and CXF

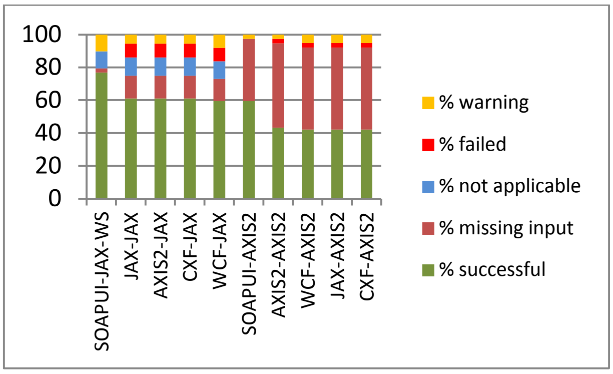

Figure 5: Conformance of SOAP messages for JAX-WS and AXIS2

5 Conclusion and Discussion

Our research involved an evaluation of four Java and .NET web service frameworks for conformance with BP1.2.

The conformance requirements are defined in very technical terms and validate the structure of the SOAP messages, WSDL documents and XSD schemas that each framework generates. They can be enabled to use additional protocols (e.g. WS-Addressing) using configuration files.

The WS-I Testing Tools Monitor and Analyzer are not easy to set up and use. Also, there is little documentation about them. If the contract-first approach is chosen to implement the web services, then each profile requirement should be manually addressed to assure conformance. The testing tools can only help us in testing conformance automatically. Some requirements are very technical and extensive, to easily consider them during the implementation of contract-first web services. If modern frameworks generate these artifacts automatically, they are conformant with BP1.2, as reported by our experiment. Based on this finding, we assumed that interoperability testing for modern frameworks is not necessary for the end user. It should be necessary for the vendors who prepare these frameworks, so we can easily develop interoperable web services using the contract-last approach. Interoperability testing is by no means an unnecessary activity.

The results of the cross-platform invocations are similar to the SoapUI invocations, with the exception of a few test cases that did not execute successfully because of some limitations of WsiHero and TcpMon that we used instead of the Monitor. WsiHero significantly simplifies the usage of the Analyzer, but the Monitor had to be substituted with a proxy server that captures all Http traffic in some cases. This flaw additionally impacted the user's experience of the testing tools.

A detailed analysis of our experiment indicated that the choice of the web service client did not impact the conformance of web services. The limitations of WsiHero resulted in some complications, but all were resolved successfully. After these resolutions, all web services were compliant with the Basic Profile 1.2.

The question is, if the profiles and testing tools are necessary for the end user today and if all modern frameworks are conformant with them. The profiles significantly contributed to the development of new versions of several WS-* standards, so their work should not be underestimated [1]. This is probably the main reason that our experiment was successful.

Our experiences with using the WS-I Testing tools were rather disappointing. Their installation and use were was tedious and not user friendly, but our experiment results confirmed the conformance of all the tested web service frameworks with the Basic Profile 1.2.

Some papers [1] [8] [9] report that certain frameworks are non conformant with BP1.0, but our experiment confirmed otherwise for BP1.2, the latest versions of frameworks and the WS-I testing tools. We also reported on our experiences with the WS-I Monitor and Analyzer (both version 1.2) for testing conformance with BP1.2.

In our further work we will also evaluate other profiles in a similar way, especially the Reliable Secure Profile 1.1 and the Basic Security Profile 1.1. We will also monitor how OASIS will continue the WS-I's work. Since the transition, no news have been reported.

References

[1] K. M. S. Kumar, S. Padmanabhuni, and A. S. Das, "WS-I Basic Profile: A Practitioner's View," *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, 2004.

[2] "WS-I, Deliverables." [Online]. Available: http://www.ws-i.org/deliverables/Default.aspx.

[3] "Basic Profile - Version 1.2 (Final)." [Online]. Available: http://ws-i.org/profiles/BasicProfile-1.2-2010-11-09.html. [Accessed: 28-Nov-2011].

[4] M. Champion, "WS-I Completes Web Services Interoperability Standards Work." [Online]. Available: http://blogs.msdn.com/b/interoperability/archive/2010/11/10/ws-i-completes-web-services-interoperability-standards-work.aspx. [Accessed: 05-Sep-2011].

[5] A. Astor and P. Yendluri, "The WS* Standards - A Primer | SOA orld Magazine." [Online]. Available: http://soa.sys-con.com/node/43974. [Accessed: 12-Jan-2012].

[6] A. Bertolino and A. Polini, "The Audition Framework for TestingWeb Services Interoperability," *31st EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 134-142, 2005.

[7] B. Simon, Z. László, B. Goldschmidt, K. Kondorosi, and P. Risztics, "Evaluation of WS-* Standards Based Interoperability of SOA Products for the Hungarian e-Government Infrastructure," *2010 Fourth International Conference on Digital Society*, pp. 118-123, Feb. 2010.

[8] S. Kuppuraju, A. Kumar, and G. P. Kumari, "Case Study to Verify the Interoperability of a Service Oriented Architecture Stack," *IEEE International Conference on Services Computing (SCC 2007)*, no. Scc, pp. 678-679, 2007.

[9] S. Papastergiou, A. Kaliontzoglou, and D. Polemi, “Interoperability Issues of a Secure Electronic Invoicing Service (Selis),” *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 3, no. 5, pp. 1-5, Sep. 2007.

[10] “WS-I, Basic Security Profile Version 1.1 Final Material.” [Online]. Available: http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html .

[11] “WS-I, Reliable Secure Profile Version 1.0 Final Material.” [Online]. Available: http://www.ws-i.org/Profiles/ReliableSecureProfile-1.0-2010-11-09.html.

[12] “Basic Profile 1.2 Test Scenario Document.” [Online]. Available: http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile.

[13] “Basic Profile Test Tools and Testing Process , Test Tools and Process Description, Final Material for Basic Profile,” 2010. [Online]. Available: http://ws-i.org/Profiles/BPTestToolsProcess-1.2-2.0-Final.zip.

[14] P. Brittenham, “Understanding the WS-I Test Tools,” *IBM Developerworks*, 2003. [Online]. Available: http://www.ibm.com/developerworks/webservices/library/ws-wsitest/.

[15] “JAX-WS Reference Implementation — Java.net.” [Online]. Available: http://jax-ws.java.net/. [Accessed: 29-Nov-2011].

[16] “Project WSIT — Java.net.” [Online]. Available: http://wsit.java.net/. [Accessed: 01-Dec-2011].

[17] “Interoperability with WCF.” [Online]. Available: http://msdn.microsoft.com/sl-si/webservicesinterop. [Accessed: 01-Dec-2011].

[18] “Apache CXF: An Open-Source Services Framework.” [Online]. Available: http://cxf.apache.org/. [Accessed: 01-Dec-2011].

[19] “Apache Axis2/Java.” [Online]. Available: http://axis.apache.org/axis2/java/core/. [Accessed: 01-Dec-2011].

[20] “soapUI - The Home of Functional Testing.” [Online]. Available: http://www.soapui.org/. [Accessed: 30-Nov-2011].

[21] “Web Services Commons : TCPMon - Apache TCPMon.” [Online]. Available: http://ws.apache.org/commons/tcpmon/. [Accessed: 30-Nov-2011].

[22] “WsiHero: A GUI utility for web services WSI validation.” [Online]. Available: http://webservices20.blogspot.com/2010/11/wsihero-gui-utility-for-web-services.html. [Accessed: 30-Nov-2011].

[23] T. Korelič and M. Heričko, “Uporabnost orodij WS-I za testiranje interoperabilnosti spletnih storitev.” technical report LIS-12-01, UM-FERI, Maribor, Slovenia, Maribor, Slovenia, 2011.

Tomaž Korelič graduated in 2009 from the Faculty of Electrical Engineering and Computer Science of the University of Maribor. His research topics include software testing, service-oriented architecture and business process management.

Marjan Heričko is a full professor for informatics at the Faculty of Electrical Engineering and Computer Science, of the University of Maribor. He is the deputy head of the Institute of Informatics and the head of the Laboratory for Information Systems. He received his Ph.D. degree in 1998 from the University of Maribor. The subject of his dissertation was quality management of object-oriented software development. His latest research includes service-oriented engineering and user-driven development.