

# Uporabnost orodij WS-I za testiranje interoperabilnosti spletnih storitev

**Tomaž Korelič, Marjan Heričko**

*Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, 2000 Maribor, Slovenija  
E-pošta: tomaz.korelic@uni-mb.si*

**Povzetek.** Interoperabilnost je ena ključnih značilnosti, ki označujejo uspešno medsebojno komuniciranje in delovanje komponent različnih interoperabilnih tehnologij. V prispevku so predstavljeni WS-I Basic Profile 1.2 (BP1.2) in testna orodja, s katerimi preverjamo interoperabilnost spletnih storitev. Izvedli smo eksperiment, v katerem smo implementirali spletne storitve v različnih tehnologijah in ogrodjih in jih prožili med seboj. Pri tem smo s pomočjo orodij WS-I za testiranje interoperabilnosti preverjali skladnost nastalih artefaktov s profilom BP1.2. V prispevku podamo izkušnje z uporabo orodij za testiranje interoperabilnosti in opišemo težave, ki smo jih imeli z uporabo orodij WS-I.

**Ključne besede:** interoperabilnost, spletne storitve, WS-I, Basic Profile 1.2, testiranje

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

The paper presents the WS-I Basic Profile 1.2 (BP1.2) and the testing tools which are used to test conformance of web services with the profile. We conducted an experiment using four web services implemented in modern frameworks. In each framework we prepared a client from which we invoked all other web services. During the execution the WS-I testing tools checked for conformance with BP1.2. The results indicate that the web services were interoperable and conformant with the profile.

## 1 UVOD

Interoperabilnost pomeni zmožnost medsebojnega komuniciranja programske in strojne opreme različnih ponudnikov, brez spoznavnih sprememb na obeh straneh [1]. Čeprav obstajajo različni standardi za izmenjevanje sporočil med spletnimi storitvami (SOAP), opis (WSDL), odkrivanje in registriranje storitev (UDDI), so ločene implementacije teh protokolov različnih ponudnikov privedle do problemov z interoperabilnostjo [1].

### 1.1 Organizacija WS-I

Organizacija Web Service Interoperability Organization (WS-I) je povzela dobre prakse pri interoperabilnosti spletnih storitev iz številnih standardov, platform, operacijskih sistemov in programskih jezikov in jih združila v t. i. profile. Ti vsebujejo implementacijske smernice o tem, kako naj bi se specifikacije spletnih storitev uporabljale, da dosežemo interoperabilnost. Med svojim delovanjem so nastali profili Basic Profile 1.1, 1.2 in 2.0, Attachments Profile 1.0, Simple SOAP Binding Profile 1.0, Reliable Secure Profile 1.0, Kerberos Token Profile 1.0, REL Token Profile 1.0,

SAML Token Profile 1.0 in Basic Security Profile 1.1 [2].

Profili vsebujejo zahteve (označene z Rxxx) in testne primere skladnosti (označene z BPxxxx, če gre npr. za Basic Profile), ki preverjajo skladnost testnih artefaktov z zahtevami. Testni primeri preverjajo statično strukturo sporočil SOAP in dokumenta WSDL. Rezultati izvedbe nakazujejo skladnost testirane spletne storitve s profilom, ne zagotavljajo pa stoddotne interoperabilnosti [3]. Testne primere izvedemo s pomočjo testnih orodij (angl. Testing tools), ki jih je prav tako zagotovila organizacija WS-I. Orodja so univerzalna in testirajo skladnost z vsemi profili. Rezultate dela organizacije WS-I so uporabili številni ponudniki spletnih storitev pri ustvarjanju novih standardov, še posebej pri SOAP 1.2 [1]. WS-I je novembra leta 2010 zaključila delo in nadaljnjo skrb za interoperabilnost prenesla na organizacijo OASIS [4].

V članku smo natančneje predstavili profil WS-I Basic Profile 1.2 in testna orodja, s katerimi preverjamo skladnost spletne storitve z njim. Predstavili smo, da večina ponudnikov tehnologij za implementacijo spletnih storitev trdi, da so skladne z Basic Profile. Izvedli smo eksperiment, v katerem smo to trditev preverili in s tem ugotovili interoperabilnost spletnih storitev. Implementirali smo spletne storitve v različnih ogrodjih in jih prožili med seboj. Pri tem smo preverjali skladnost artefaktov, ki pri tem nastanejo z Basic Profile 1.2, z uporabo WS-I testnih orodij (WS-I testing tools). Poleg tega smo ugotovili, ali so spletne storitve dejansko interoperabilne. Cilj raziskave je ugotoviti uporabnost orodij WS-I za testiranje skladnosti s profili in podati izkušnje z njihovo uporabo. Želimo ugotoviti, ali so rezultati, ki jih orodja podajo pri testiranju

interoperabilnosti spletnih storitev, uporabni za načrtovalca spletnih storitev.

### 1.2 Standardi WS-\*

Interoperabilnost spletnih storitev rešujejo in zagotavljajo t. i. standardi WS-\*. Za pravilen transport sporočil skrbijo HTTP, HTTPS in SMTP. Na ravni sporočil so definirani standardi XML (vključno z XSL, Xpath idr.), SOAP, *Attachments for SOAP* in ws-Addressing, ki spletnim storitvam omogoča izmenjavo naslovnih informacij. Za opis storitev skrbijo WSDL, UDDI, WS-Inpection, WS-Inspection in WS-MetadataExchange. WS-ReliableMessaging je namenjen zanesljivi dostavi sporočil, WS-Security varni komunikaciji, WS-Transaction izvajanju transakcij in WS-BPEL kompoziciji storitev. za pravilno izvajanje transakcij [5]. Čeprav so se ti standardi dobro prijeli v praksi, prihaja do težav pri interoperabilnosti zaradi lastnih implementacij ogrodij ponudnikov spletnih storitev. Profili WS-I kot osnovo uporabljajo večino teh standardov, z izjemo WS-Transaction in WS-BPEL, in testirajo skladnost nastalih artefaktov z njimi.

## 2 SORODNA DELA

Kumar et.al. [1] so predstavili ključne probleme interoperabilnosti, ki so nastali pri uporabi v praksi, in analizirajo, kako jih naslavlja Basic Profile 1.0 (BP). Omejitve profila so evalvirali z najpopularnejšimi implementacijami spletnih storitev. Poudarjajo omejitve, da BP ne naslavlja problemov interoperabilnosti podatkovnih tipov. Dodatne težave so odkrili pri podatkovnih tipih v XML shemah v SOAP sporočilu, glavi SOAPAction in razširljivosti elementov v WSDL1.1. Njihove ugotovitve se nanašajo na BP1.0 iz leta 2004, v naši raziskavi pa bomo uporabili aktualno verzijo BP1.2 iz leta 2010. Uporabili so ista orodja za testiranje skladnosti kot mi, s katerimi so preverili skladnost ogrodja Apache AXIS 1.0. Ugotovili so, da sporočila SOAP in WSDL, ki jih je AXIS 1.0 generiral, niso bila skladna s priporočili BP1.0. V naši raziskavi smo njihov eksperiment ponovili z uporabo BP1.2 in AXIS2 1.6.1 in ugotovili, da so bile naše spletne storitve skladne s profilom.

Bertolino et.al. [6] so predlagali novo ogrodje za testiranje spletnih storitev, ki temelji na diagramih stanj (protocol state machine diagrams). Ogrodje preverja zaporedje proženj med dvema različnima storitvama in pri tem odkriva težave pri interoperabilnosti. Avtorji v nasprotju z našo raziskavo ne uporabljajo orodij WS-I. Predlagana rešitev uspešno rešuje težave z interoperabilnostjo z uporabo novih pristopov.

Simon et.al. [7] so pripravili nabor testnih primerov, ki ocenjujejo interoperabilnost produktov SOA na podlagi standardov WS-\*. Testni primeri so ponovno izvedljivi, testno okolje pa je dovolj fleksibilno, da lahko vanj vključimo nove storitve in jih testiramo z obstoječimi. Rezultate testov so evalvirali na produktih različnih pomembnih ponudnikov ogrodij spletnih storitev. Ogrodje s pomočjo avtomatskih testnih

primerov uspešno preverja skladnost spletne storitve z različnimi standardi WS-\*.

Kuppuraju et.al. [8] so vzpostavili novo metodologijo, s katero so preverili interoperabilnost različnih produktov, ki so jih izdelali pri študiji primera. Za svoje storitve so uspešno uporabili testna orodja WS-I, da bi zagotovili osnovne zahteve interoperabilnosti. Podrobnih izkušenj pri uporabi profilov ali orodij ne podajajo.

Papastergiou et.al [9] predstavljajo nov e-računovodski sistem, ki dosega visoko stopnjo interoperabilnosti, saj uspešno prestane večino testnih primerov skladnosti profilov WS-I. V njihovem primeru so uspešno uporabili testna orodja WS-I. Dodatno opozarjajo, da profili WS-I ne pokrivajo problemov interoperabilnosti v celoti, ker v glavnem naslavlajo sporočila med storitvami. Pri tem poudarjajo probleme pri digitalnem podpisovanju sporočil, saj je njihov sistem temeljil na uporabi različnih ogrodij za digitalno podpisovanje. Identificirali so srednjo do visoko interoperabilnost njihovih spletnih storitev na podlagi testiranja z BP1.0.

Odkrili smo dve vrsti študij, ki testirajo interoperabilnost spletnih storitev.

V prvo skupino spadajo študije, ki uporabljajo orodja za testiranje interoperabilnosti organizacije WS-I [1] [8] [9]. Nekateri avtorji omenjajo težave in pomanjkljivosti pri rezultatih testiranja in pri neskladnosti s profili. Naša raziskava se razlikuje v uporabljeni verziji profilov in ogrodij spletnih storitev. V drugi skupini so študije, ki vpeljujejo nova ogrodja [6] [7] in se v tem pogledu razlikujejo od naše raziskave. Po njih smo se zgledovali pri pripravi testnih primerov. Naša raziskava poleg rezultatov skladnosti dodaja izkušnje pri uporabi orodij WS-I za testiranje interoperabilnosti.

## 3 PROFILI WS-I

Orodja WS-I za testiranje interoperabilnosti s testnimi primeri preverjajo skladnost dokumenta WSDL, sporočil SOAP in XML shem s profili. V naši raziskavi se bomo omejili na Basic Profile 1.2 [3], obstajajo pa še drugi, ki preverjajo interoperabilnost z WS-Reliable Messaging, WS-Security in drugo. WS-I je izdal tudi Basic Profile 2.0, ki vsebuje enake zahteve, le da preverja skladnost s Soap1.2, BP1.2 pa Soap1.1. Ker smo se v raziskavi odločili za uporabo Soap1.1 pri testnih spletnih storitvah, smo uporabili BP 1.2 [3].

### 3.1 Struktura profila

Vsak profil je zgrajen na podoben način. V formatu XML so predstavljene zahteve profila, ločen TAD dokument (Test Assertion Document) pa vsebuje testne primere, ki s pomočjo izrazov Xpath preverjajo skladnost z zahtevami [3][10][11]. BP1.2 in BP2.0 imata dokument TAD že vgrajen v profil. V specifikacijah profila je eksplicitno navedeno, da ne zagotavljajo interoperabilnosti, naslavlja le pogoste probleme, ki so jih odkrili iz prakse.

Prvo poglavje profila Basic Profile 1.2 opisuje svoje splošne lastnosti, drugo pa konkretno navede, kaj in kako testira. Nadaljnja poglavja predstavljajo zahteve in testne primere, ki preverjajo skladnost testnih artefaktov z njimi. Naslednje točke predstavljajo nekatere zahteve iz BP1.2:

**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 Testni primeri skladnosti BP1.2

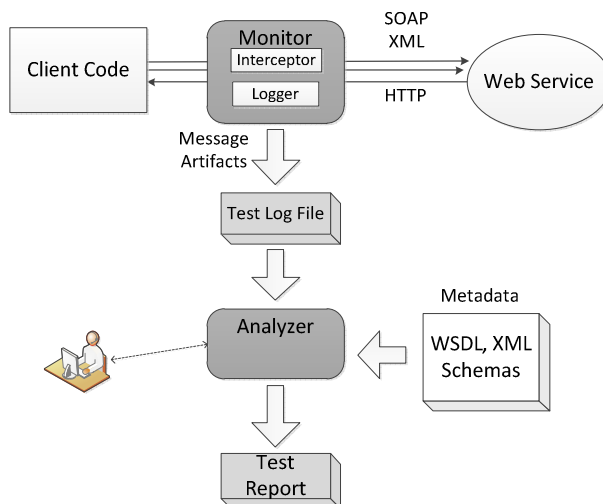
Zahteve skladnosti v sporočilih SOAP, dokumentih WSDL in XML shemah preverjajo pravilno strukturo Soap sporočil, natančneje, uporabo HTTP/1.1, serializacija sporočil, strukturo ovojnice SOAP, napake SOAP, uporabo SOAP v HTTP, WS-Addressing, v WSDL pa elemente *portType*, *Messages*, *types*, *SOAP Binding* in metapodatke WS-Addressing.

BP1.2 naslavlja uporabo SOAP 1.1 prek HTTP1.1, WS-Addressing 1.0, serializacijo sporočila, ovojnice SOAP, SOAP napake in uporabo URI. V WSDL preverja strukturo dokumenta, podatkovne tipe (samo *Array* in *anyType*), *portTypes*, *Binding*, *XML Sheme* in *WS-Addressing*. Poglavji 5 in 6 preverjata UDDI in uporabo HTTPS, vendar nimata pripadajočih testnih primerov. Pomembno je, da je le 66 % zahtev pokritih s testnimi primeri skladnosti [12].

### 3.3 Orodja WS-I za testiranje skladnosti

WS-I je razvil dve orodji, ki sta testno povezani pri testiranju skladnosti s profili – Monitor in Analyzer. Trenutna verzija 1.2 (in 2.0 za BP2.0) testnih orodij se bistveno razlikuje pri vzpostavitvi in delovanju, vendar pa osnovni koncept ostaja enak. Nova verzija monitorja za delovanje potrebuje spletni strežnik Apache, Python in Perl, medtem ko se je stara pognala zgolj iz ukazne vrstice.

Monitor je komponenta, ki deluje kot vmesni člen (*man in the middle*) in prestreza sporočila SOAP, ki si jih klient in spletna storitev pošiljata prek protokola HTTP. Z vsemi zajetimi sporočili orodje skupaj z dokumentom WSDL in XML shemami izdela log datoteko (Test Log file). V drugem koraku komponenta Analyzer log datoteko in druge dokumente (WSDL in XML sheme) analizira in nad njo izvede testne primere skladnosti. Rezultat uspešnosti testiranja se prikaže v poročilu testiranja (Test Report) [13]. Slika 1 prikazuje shemo delovanja orodij WS-I za testiranje interoperabilnosti.



Slika 1: Shema delovanja orodij za testiranje WS-I

Poročilo prikaže testne primere skladnosti, ki so se izvedli nad artefakti, in njihove izide. Izidi testov skladnosti so lahko uspešno (»Successful«), neuspešno (»Failed«), ni relevantno (»Not relevant«), manjka vhod (»Missing input«) in opozorilo (»Warning«). Če med njimi ni niti enega, ki se je izvedel s statusom neuspešno, ocenimo, da je spletna storitev skladna s profilom [14].

## 4 EKSPERIMENT

Z eksperimentom smo želeli preveriti, ali so ogrodja spletnih storitev različnih ponudnikov skladna s profilom BP1.2 in posledično interoperabilna. Izbrali smo štiri ogrodja spletnih storitev, v katerih smo implementirali testne spletne storitve. Načrtovali smo testne primere, s katerimi smo prožili spletne storitve, in pri tem preverjali skladnost z BP1.2 z uporabo »WS-I testing tools«. Prožili smo jih iz orodja SoapUI in medsebojno med ogrodji. S tem smo preverili interoperabilnost spletnih storitev. Pri tem smo pridobili izkušnje z vzpostavitvijo in uporabo orodij za testiranje interoperabilnosti, kar je bil tudi glavni cilj naše raziskave.

Številni ponudniki ogrodiv spletnih storitev uradno podpirajo skladnost s profili WS-I. Projekt JAX-WS predstavlja referenčno implementacijo SOAP XML spletnih storitev (XML Web Services JAX-WS) in je jedro projekta Metro znotraj skupnosti Glassfish. Uradno JAX-WS podpira Basic Profile 1.2 in 2.0 [15]. Znotraj projekta Metro obstaja projekt WSIT (Web Services Interoperability Technologies), ki vključuje implementacijo tehnologij WS-Trust, WS-SecureConversation, WS-SecurityPolicy, WS-Reliable Messaging, WS-AtomicTransactions/ Coordination, WS-MetadataExchange in SOAP prek protokola TCP.

Po uradni izjavi je celoten projekt Metro testiran za interoperabilnost med platformo Java 6 in Windows Communication Foundation (WCF) na platformi .NET 3.0 in 3.5 [16].

Microsoft za svojo tehnologijo WCF WS-\* znotraj .NET 4 trdi, da je skladna z vsemi profili WS-I in interoperabilna z največjimi ponudniki spletnih storitev (Oracle, IBM, SAP, Project Metro) [17]. Vse nastavitve v zvezi z interoperabilnostjo lahko nastavimo v datoteki *web.config* znotraj projekta WCF v orodju Visual Studio 2010.

Ogrodje CXF implementira JAX-WS API. Na uradnih spletnih straneh je navedeno, da je skladno z Basic Profile 1.1, podpira pa WS-Addressing, Soap 1.1 in Soap 1.2 [18].

V nasprotju s prejšnjimi tehnologijami uradna spletna stran ogrodja AXIS2 ne vsebuje informacije o skladnosti s profili WS-I [19]. Omenja zgolj podprte specifikacije SOAP1.1 in 1.2, WSDL 1.1 (SOAP in http Binding), WS-Addressing, WS-Policy in transport prek http, smtp, jms in tcp. Na podlagi teh podprtih tehnologij bi lahko predvidevali, da bo tudi AXIS2 skladen z Basic Profile 1.2.

Večina ponudnikov tehnologij spletnih storitev je po lastnih izjavah skladna z WS-I Basic Profile 1.0 ali 1.2. Z eksperimentom bomo ugotovili, ali so posledično interoperabilne, in podali izkušnje pri uporabi orodij WS-I.

#### 4.1 Testne spletne storitve

Izbrali smo štiri ogrodja za implementacijo spletnih storitev: WCF 4, Jax-WS 2.2.5, CXF 2.4. in Axis2 1.6.1. Implementirali smo funkcionalnosti, s katerimi lahko najbolje pokrijemo vse artefakte, ki jih testira Basic Profile 1.2. Tabela 1 prikazuje podrobne informacije uporabljenih ogrodij skupaj z aplikacijskim strežnikom, ogroddjem IDE, programskim jezikom in načinom konfiguracije. Pri implementaciji storitev se je dokument WSDL v vseh ogroddjih generiral samodejno po principu *contract-last*.

Tabela 1: Pregled uporabljenih ogrodij

WS-API in Ime ponudnika	Aplikacijski strežnik	IDE	Programski jezik	Konfiguracija
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 Testni primeri

Testne primere smo načrtovali po zgledu raziskave Simon et.al. [7]. Avtorji so v svoji raziskavi testirali

interoperabilnost standardov WS-\* s pomočjo svojih testnih primerov, brez profilov WS-I. Kljub temu so testni primeri primerni za preverjanje skladnosti z njimi, saj njihovo proženje kreira številne artefakte in strukturne elemente, ki so del sporočil SOAP. Če se držimo njihovih napotkov in pripravimo podobne spletne storitve, bodo ogrodja kreirala primeren dokument WSDL in sporočila SOAP, nad katerimi bomo lahko preverjali skladnost z WS-I Basic Profile 1.2 (BP1.2). V poglavju 3.2 smo predstavili specifikacije Basic Profile 1.2 in artefakte, ki jih vključuje. Z analizo teh artefaktov in na podlagi priporočil Simon et.al. [7] smo pokrili večino artefaktov, ki jih preverja BP1.2. Testni primeri skladnosti profilov so zelo tehnično zasnovani in validirajo strukturo testnih artefaktov. Ogrodja spletnih storitev vse te artefakte generirajo samodejno, zato nismo pripravljali samostojnih testnih primerov za vsak testni primer skladnosti posebej. V raziskavi smo se omejili na SOAP 1.1 prek HTTP, WS-Addressing, obvladovanje napak in uporabo podatkovnih tipov, ki jih preverja BP1.2. Tehnologij MTOM, SOAP 1.2 in uporabe HTTPS nismo vključili v raziskavo. Pri vseh testnih spletnih storitvah in odjemalcih smo vključili obvezno uporabo WS-Addressing 2005-08.

Basic profile 1.2 ne preverja uporabljenih podatkovnih tipov. Kljub temu smo v enem testnem primeru uporabili podatkovni tip *Date*. S tem želimo ugotoviti, kako posamezne tehnologije pretvarjajo ta podatkovni tip med seboj. Poglavje 4.3 v specifikacijah Basic Profile 1.2 omenja, da BP testira strukturo sporočil SOAP, ki vsebujejo podatkovni tip *List* [3]. Zato smo ga v en testni primer vključili (*ArrayList<String>*).

Če povzamemo, smo v štirih različnih ogroddjih pripravili po eno spletno storitev s sedmimi operacijami. Naslednje testne primere smo izvedli samo s proženjem štirih operacij:

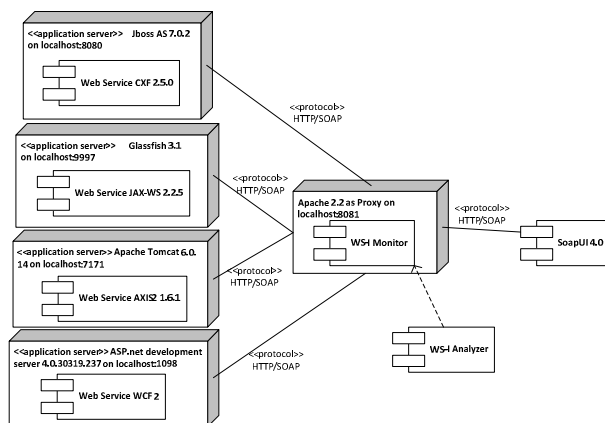
- Proženje operacije *float Divide(54.3, 3.0)* z WS-Addressing 2005-08.
- Proženje operacije *int Divide(Integer(54, 0)* z WS-Addressing 2005-08. Pričakovan rezultat proženje izjeme (*Divide with 0*).
- Proženje operacije *Date GetCurrentDate()* z WS-Addressing 2005-08. Pričakovan rezultat izpis trenutnega datuma.
- Proženje operacije *ArrayList<String> GetEventList(Date)* z WS-Addressing 2005-08. Pričakovan rezultat izpis seznama dogodkov velikosti 3 na določen datum (5.11.2011).

#### 4.3 Uporabljena orodja

V eksperimentu smo za testiranje skladnosti uporabili orodja Monitor 1.2, Analyzer 1.2, TCPMon 1.0, WSIHero 1.0 in SoapUI 4.0. Spletne storitve in odjemalce smo implementirali v orodjih IDE Visual Studio 2010, Netbeans 7.0 in Eclipse Indigo.

#### 4.4 Izvedba

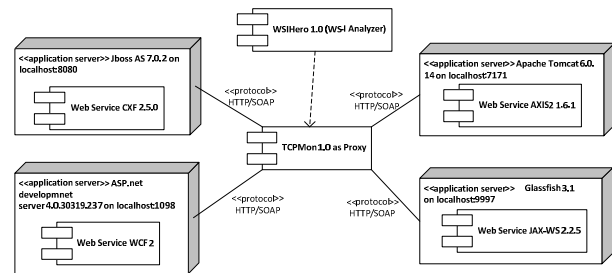
V prvem koraku eksperimenta smo prožili storitve iz orodja SoapUI 4.0 [20]. Vsa proženja so se izvedla prek komponente monitor, ki je bil vzpostavljen kot namestniški (proxy) strežnik na vratih 8181. SoapUI omogoča s svojimi nastavitvami proženje testnih spletnih storitev prek namestniškega (proxy) strežnika. Testne spletne storitve so bile nameščene na lastnih aplikacijskih strežnikih. Po proženju posamezne storitve je monitor kreiral log datoteko. V drugem koraku izvedbe vseh testnih primerov smo pognali še WS-I Analyzer, ki je na podlagi poročila iz komponente Monitor in dokumenta storitve WSDL kreiral končno poročilo skladnosti z WS-I Basic Profile 1.2. Slika 2 prikazuje testno okolje s štirimi aplikacijskimi strežniki, ki gostijo spletne storitve in komponento SOAPUI, ki prek namestniškega (proxy) strežnika (monitor) proži storitve. V drugem koraku smo prožili storitve med seboj. V vseh štirih ogroddjih smo pripravili odjemalce, iz katerih smo prožili vse tri storitve iz drugih ogroddij in samo sebe (slika 3).



Slika 2: Shema prvega sklopa scenarijev

V drugem koraku smo zaradi tehničnih težav pri vzpostavitvi orodja *monitor* kot namestniškega steznika (proxy) iz okolij IDE kot alternativo uporabili orodje Tcpmon 1.0 [21] in WsiHero 1.0 [22]. Orodje Monitor ni prestrezalo sporočil SOAP, čeprav smo v odjemalcih spletnih storitev nastavili proxy ustrezni strežnik. Komponenta TcpMon je zamenjala Monitor in je delovala kot namestniški (proxy) strežnik, ki je prestrezal celoten promet med odjemalci in storitvami. Iz tega smo izluščili sporočila SOAP, ki smo jih kot vhod podali v WsiHero [22]. Orodje s pomočjo grafičnega uporabniškega vmesnika bistveno olajša postopek analiziranja skladnosti artefaktov s profili WS-I. Pri tem uporablja orodje WS-I Analyzer v ozadju, pri tem pa je uporabniku bolj prijazno za uporabo kot sam analyzer. Žal ima WsiHero določene pomanjkljivosti. Orodje prejme sporočilo SOAP v formatu XML brez pripadajoče http glave (*header*) in ker nekateri testni primeri skladnosti preverjajo tudi glavo HTTP, so ti bili

posledično neuspešni in so opozarjali na morebitno odsotnost glav HTTP.



Slika 3: Shema prvega sklopa scenarijev

#### 4.5 Rezultati

Orodje Analyzer prikaže rezultate v obliki poročila, ki je razdeljeno na dva dela. Prvi vsebuje testne primere, ki so se izvedli nad izmenjanimi sporočili Soap, drugi pa nad WSDL in shemami. Opis storitve (WSDL) je vedno enak, zato so rezultati testnih primerov skladnosti nad njim posledično vedno enaki pri vseh proženjih storitve.

Vsi testni primeri nad dokumenti WSDL so se v vseh primerih izvedli uspešno. Vsi dokumenti WSDL testnih spletnih storitev so bili v celoti skladni z Basic Profile 1.2.

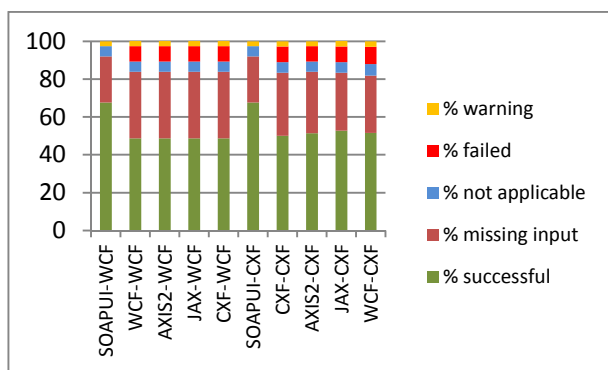
Sliki 4 in 5 prikazujeta rezultate izvedbe testnih primerov skladnosti za sporočila SOAP. Kot opazimo, se je večina testnih primerov skladnosti izvedla uspešno, če smo storitve prožili iz SoapUI, noben pa neuspešno.

Vsem ogroddjem sta skupna testna primera skladnosti BP1901 ali BP1905, ki sta vrnila opozorilo. Nanaša se na opsijsko uporabo dodatnega elementa pri priklicu izjeme, ki jo WS-I odsvetuje. Storitve ogroddja JAX-WS je vrnila dve dodatni opozorili BP1090a in BP1153a. BP1090a se nanaša na wsa:Action element v sporočilu SOAP, ki je vrnilo napako. Element je zavzel vrednost <http://www.w3.org/2005/08/addressing/fault>. BP1153a opozarja na morebitno odsotnost elementa wsa:To, vendar smo po pregledu sporočila ugotovili, da je bil element prisoten.

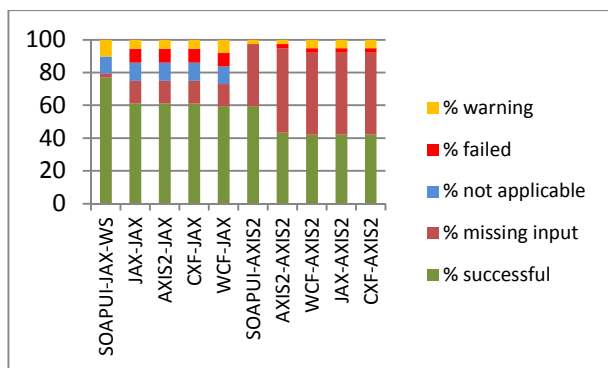
Kot opazimo iz slik, so bili rezultati proženja storitve JAX-WS najuspešnejši, saj se je uspešno izvedel največji delež testnih primerov skladnosti. Po drugi strani po smo pri tem proženju pridobili največje število opozoril (skupaj 4). Ker smo se pri načrtovanju testnih primerov zgledovali po poglavjih profila BP1.2 in ne po vsaki zahtevi posebej, se nekateri testni primeri skladnosti niso izvedli ali pa niso bili relevantni. Končali so se z izidom »manjka vhod« (»missing input«), kar je obarvano s temno rdečo barvo. Tak rezultat nastane, če primarni tip podatka ni bil specificiran kot vhod v orodje Analyzer ali ni obstajal v podanem testnem artefaktu. Storitve AXIS2 ima največji delež takšnih izidov.

Rezultati proženja iz SoapUI bodo osnovno merilo za primerjavo s proženji iz drugih ogroddij. Vsa proženja iz orodja SoapUI so bila uspešna in skladna z BP1.2.

Pri proženjih med tehnologijami opazimo manjši delež neuspešnih testnih primerov skladnosti. Pri tem gre za testne primere skladnosti BP1126a, BP1143c, BP1144, ki so padli zaradi odsotnosti glave HTTP, kar pripisujemo omejitvi orodja WsiHero (predstavljene v poglavju 4.4). Dodatno se zaradi istega razloga niso izvedli BP1007, BP1019, BP1153a, BP1208 in BP1600. Natančnejše rezultate eksperimenta smo predstavili v svojem tehničnem poročilu [23].



Slika 4: Testi skladnosti SOAP sporočil WCF in CXF



Slika 5: Testi skladnosti SOAP sporočil JAX-WS in AXIS2

## 5 SKLEP IN RAZPRAVA

V raziskavi smo preverili skladnost štirih ogrodij spletnih storitev Java in .NET z BP1.2. Izvedli smo štiri sklope testnih scenarijev, ki so najbolje pokrili večino testnih primerov skladnosti iz BP1.2.

Zahteve skladnosti BP1.2 so napisane zelo tehnično in se nanašajo na strukturo sporočil SOAP, WSDL in XML shem, ki so odvisni od ogrodja, v katerih implementiramo spletne storitve. Ogrodja uporabniku s pomočjo nastavitvenih datotek omogočajo vklop dodatnih protokolov (npr. WS-addressing) in ustrezno vsebino v sporočilih SOAP in WSDL.

Orodji Monitor in Analyzer sta uporabniku neprijazni za vzpostavitev in uporabo. Dodatno obstaja o njih zelo malo dokumentacije. Če bi pri implementaciji spletnih storitev uporabili pristop *contract-first*, bi morali zahteve profilov natančneje proučiti in jih upoštevati pri implementaciji storitev, orodja za testiranje skladnosti pa bi nam pomagala pri avtomatskem preverjanju

skladnosti. Zahteve skladnosti so zelo tehnične in deloma obsežne, kar zna delo pri tem otežiti. Naša raziskava je pokazala, da so spletne storitve implementirane s samodejnim generiranjem dokumenta WSDL skladne z BP1.2. Na podlagi te ugotovitve in rezultatov lahko sklepamo, da uporaba orodij za testiranje skladnosti ni potrebna. Ker se pri principu *contract-last* artefakti generirajo samodejno, so profili in orodja za testiranje pomembnejši za ponudnike ogrodij. Preverjanje skladnosti s profili zanesljivo ni odveč.

Pri medsebojem proženju so rezultati zelo podobni, proženji iz SoapUI, z izjemo določenih testnih primerov skladnosti, ki se ne izvedejo uspešno zaradi omejitev orodja WsiHero, ki smo ga uporabili zaradi neuspešnega prestrezanja sporočil SOAP komponente Monitor pri medsebojnih proženjih. WsiHero olajša delo z orodjem Analyzer, Monitor pa lahko nadomestimo s poljubnim namestniškim strežnikom, ki zajema promet sporočil SOAP. Ta slabost je dodatno poslabšala uporabniško izkušnjo z orodji.

Po natančnem pregledu vseh proženj smo ugotovili, da izbira odjemalca iz ogrodij ali SoapUI ni bistveno vplivala na rezultate skladnosti. Omejitve orodja WsiHero so pripeljale do manjših razlik med proženji, vendar smo lahko vse razlike pojasnili in obrazložili.

Po analizi med rezultati ni bilo niti enega testnega primera skladnosti, ki se ne bi izvedel neuspešno.

Pri tem se zastavi vprašanje uporabnosti profilov in orodij za testiranje skladnosti, ker smo ugotovili, da so spletne storitve skladne z njimi. Profili so veliko prispevali pri izdelavi novih verzij standardov WS-\* [1]. Morda je bil naš eksperiment ravno zato tako uspešen.

Izkušnje z uporabo orodij WS-I za testiranje interoperabilnosti so bile mešane. Vzpostavitev in uporaba orodij sta prinesli kar nekaj težav, rezultati eksperimenta pa so potrdili interoperabilnost.

Sorodna dela [1] [8] [9] opozarjajo na neskladnosti profila BP starejše verzije. Naša raziskava teh pomanjkljivosti ni identificirala, kar lahko pripišemo uporabi finalne verzije profila, orodij za testiranje interoperabilnosti in ogrodij spletnih storitev. V nasprotju z drugimi sorodnimi deli, smo podali izkušnje z uporabo orodij Monitor in Analyzer (verzije 1.2) v povezavi z Basic Profile 1.2.

Kot nadaljnje delo bi lahko podobno ovrednotili tudi druge profile, predvsem Reliable Secure Profile 1.1 in Basic Security Profile 1.1. Zanimivo bo spremljati, kaj bo kot prvi korak storila organizacija OASIS, ki je prevzela delo WS-I. Od prevzema novembra leta 2010 do danes (januar 2012) na uradnih spletnih straneh ni bilo nobene posodobitve ali novice.



## LITERATURA

- [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 World Magazine." [Online]. Available: <http://soa.sys-con.com/node/43974>. [Accessed: 12-Jan-2012].
- [6] A. Bertolino and A. Polini, "The Audition Framework for Testing Web 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. Sec, 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://webservicess20.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** je diplomiral leta 2009 na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru. Med njegova področja raziskovanja spada testiranje programske opreme, SOA in upravljanje poslovnih procesov.

**Marjan Heričko** je redni profesor za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru. Je namestnik predstojnika Inštituta za informatiko in vodja Laboratorija za informacijske sisteme. Doktoriral je leta 1998 na Univerzi v Mariboru na področju zagotavljanja kakovosti objektno-orientiranega razvoja programske opreme. Njegovo raziskovalno delo zajema vsa področja razvoja sodobnih informacijskih rešitev in storitev s poudarkom na naprednih pristopih k modeliranju in načrtovanju informacijskih sistemov, načrtovalskih vzorcih in metrikah. V zadnjem času je aktiven tudi na področju storitvene znanosti in storitvenega inženirstva, kot tudi uporabniško usmerjenega razvoja.