

Semantički web servisi

Inicijative za uvođenje semantike u web servise

Velibor Radulović
Odjeljenje za inf. tehnologije
Investiciono-razvojna banka RS
Banja Luka, Bosna i Hercegovina
velibor.radulovic@irbrs.org

Čedomir Radulović
Služba za informacijski sistem
Zavod za izgradnju Banjaluka
Banja Luka, Bosna i Hercegovina
cedomir.radulovic@gmail.com

Zoran Đurić
Katedra za računarsku tehniku
Elektrotehnički fakultet
Banja Luka, Bosna i Hercegovina
zoran.djuric@etfbl.net

Sadržaj—Tehnologije web servisa bave se uglavnom sintaksnim aspektima web servisa i njihovo prilagođavanje dinamičkom okruženju kakav je savremeni WWW, je gotovo nemoguće bez intervencije čovjeka. Inicijative za uvođenje semantike u web servise pojavile su se s ciljem dopunjavanja međudjelovanja klasičnih web servisa, kako bi se omogućio rad u okruženjima sa heterogenim podacima i različitim ponašanjima. U ovom radu predstavljene su tri inicijative semantičkih web servisa. Analiziranjem njihovih različitih koncepata sagledani su pravci razvoja tehnologije semantičkih web servisa i procijenjene mogućnosti u današnje vrijeme.

Ključne riječi—web servis; semantički web; owl-s; wsmo; meteor-s; ontologija;

I. UVOD

Tehnologije semantičkog web-a omogućuju da informacije na web-u postanu „razumljive“ računarima, što znači da mogu biti procesirane i rezonovane od strane računarskih algoritama sa većim stepenom automatizma u odnosu na dosadašnji, sintakсни web, gdje su se informacije uglavnom pojavljivale u formi razumljivoj samo ljudima.

Sam koncept računarski razumljivih informacija u semantičkom webu ne implicira bilo kakvu vrstu vještačke inteligencije ili obrade ljudskog jezika. Semantički web samo naglašava mogućnost računara da rješava dobro definisane probleme pomoću dobro definisanih operacija na postojećim dobro definisanim podacima. Ideja je oblikovati formalizam kojim se omogućuje opis informacije na webu na način prikladan računarskoj obradi.

Ontologija kao ključna komponenta semantičkog web-a, predstavlja formalizovani rječnik termina koji definiše koncepte i relacije za opisivanje i prezentovanje određene oblasti interesovanja, i koji je djeljiv od strane članova zajednice. Glavna ideja je da se postigne zajedničko shvatanje strukture informacija među ljudima i/ili računarima, da se pretpostavke unutar domena eksplicitno definišu, te da se jednom zapisano znanje može ponovo upotrijebiti za iste ili

različite primjena. U semantičkom webu ontologije imaju ključnu ulogu – definisanje značenja pojmova (resursa) na web-u. Za kreiranje ontologija koriste se standardni ontološki jezici poput OWL (*Web Ontology Language*).

Posljednjih godina težilo se uspostavljanju standarda za razvoj i upotrebu resursa weba kroz ono što će postati poznato kao web servisi. Jedna od definicija web servisa je: „Web servis je softverski sistem dizajniran da podrži interoperabilnu interakciju između mašina putem mreže. Posjeduje interfejs opisan formatom koji mašine mogu obrađivati (posebno WSDL – *Web Service Description Language*). Drugi sistemi stupaju u interakciju sa web servisom na način koji je definisan njegovim opisom upotrebom SOAP (*Simple Object Access Protocol*) poruka, obično prenošenih HTTP (*HyperText Transfer Protocol*) protokolom sa XML (*Extensible Markup Language*) serijalizacijom zajedno sa drugim web standardima.“ [1]

Web servisi su dodali novi nivo funkcionalnosti postojećem web-u, otvarajući put prema lakšoj integraciji distribuiranih komponenti. Ipak, klasične tehnologije web servisa bave se samo sintaksnim aspektima web servisa i stoga omogućuju skup rigidnih servisa koji se ne mogu prilagoditi na promjenljivo okruženje bez intervencije čovjeka.

Vizija semantičkih web servisa je kombinacija ove dvije tehnologije, kako bi se omogućila automatska, dinamička interakcija među softverskim sistemima. Tehnologije klasičnih web servisa omogućuju opisivanje interfejsa na standardizovan način, ali ne govore ništa o tome šta softverski sistem radi, ili kakva sekvenca poruka se koristi za interakciju s njim. Ovo se može prevazići korištenjem tehnologija semantičkog web-a. Tako, na primjer, ontologijama možemo kreirati opise ponuđenih servisa, koji mogu biti interpretirani od strane računara, što nam dalje pruža šansu za sofisticiranije metode pronalazjenja servisa, u odnosu na ono što je trenutno moguće sa UDDI-jem (*Universal Description, Discovery and Integration*).

Kombinovanje ovih tehnologija moguće je, na primjer, pronaći određenu uslugu preko Interneta, locirati njenog provajdera, dogovoriti određenu poslovnu saradnju i plaćanjem dobiti pristup servisu i funkcionalnosti koje on nudi, a sve to potpuno automatizovano i veoma brzo. Takođe, moguća bi bila situacija da se kombinovanjem više servisa stvori kompleksniji servis. Pored toga, u slučaju kada jedna od komponenti postane nedostupna, zamjena bi se mogla brzo pronaći, tako da kompleksni servis može i dalje funkcionisati.

Osnovna motivacija ovog rada je da se proučavanjem različitih koncepata uvođenja semantike u web servise sagledaju pravci razvoja tehnologije semantičkih web servisa i procijene njene mogućnosti u današnje vrijeme.

II. INICIJATIVE SEMANTIČKIH WEB SERVISA

Inicijative za uvođenje semantike u web servise pojavile su se s ciljem dopunjavanja međudjelovanja klasičnih web servisa, kako bi se omogućio rad u okruženjima sa heterogenim podacima i različitim ponašanjima servisa, te omogućila automatizacija pri otkrivanju servisa, kao i dinamička kompozicija i pozivanje servisa. Osnovni princip ovih inicijativa je proširenje sintaksnih opisa servisa semantičkim slojem kojeg računari mogu interpretirati i na određeni način rezonovati. Ontologije igraju centralnu ulogu u definisanju semantičkog proširenja.

U ovom radu biće predstavljene tri inicijative semantičkih web servisa: OWL-S, WSMO i METEOR-S.

A. OWL-S

OWL-S predstavlja skup ontologija namijenjenih semantičkom opisivanju web servisa. Jezik OWL korišten je s ciljem da se korisnicima i softverskim agentima omogući da, sa visokim stepenom automatizma, otkrivaju, pozivaju, vrše kompoziciju i prate resurse web-a koji nude određene servise i imaju određene osobine.

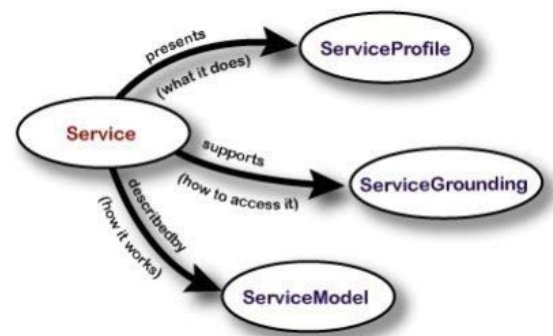
OWL-S kombinuje ekspresivnost deskriptivne logike, kakvu ima OWL, sa pragmatizmom postojećih standarda web servisa, kako bi se servisi semantički opisali i kako bi im se moglo pristupiti kroz dobro definisan formalizam tipova podataka. To konkretno znači da OWL-S standardima web servisa, kakvi su SOAP, WSDL, UDDI i BPEL4WS (*Business Process Execution Language for Web Services*), dodaje bogatije tipove i klase informacija kako bi se na znatno efikasniji način opisale mogućnosti web servisa, u odnosu na to kako se isto može uraditi standardnim XML tipovima podataka. Pored toga, reprezentacije klasa se integrišu sa procesnim modelom, tako da se ne prati samo tok podataka i upravljačkih informacija, već i njihovi prateći efekti i preduslovi. Upotreba ovakvog jezika omogućuje grupisanje servisa i tipova podataka u taksonomske hijerarhije, zajedno sa definicijama relacija i ograničenjima među klasama i njihovim instancama. Dobro definisana semantika omogućuje automatsku manipulaciju ovih struktura.

Glavni aspekti ove ontologije su:

- Servis (*Service*) koja služi kao organizaciona referentna tačka za deklarisanje web servisa. Svaki servis je deklarisan kreiranjem instance *Service* klase,

tj. za svaki objavljeni servis postoji jedna instanca *Service* klase.

- Servisni profil (*ServiceProfile*) koja opisuje šta servis radi, opisujući njegovu funkcionalnost i ostala nefunkcionalna svojstva koja su korištena za lociranje servisa na bazi njihovog semantičkog opisa.
- Model servisa (*ServiceModel*) koja opisuje kako servis radi, odnosno kako postiže svoju funkcionalnost, uključujući detaljne opise njegovih konstitutivnih procesa.
- Pristup servisu (*ServiceGrounding*) koja opisuje kako koristiti servis, odnosno kako se servisu može pristupiti.



Slika 1. Grafička reprezentacija glavnih klasa OWL-S ontologije

Kako je prikazano na Slici 1, klasa *Service* se vezuje za svoje podklase *ServiceProfile*, *ServiceModel* i *ServiceGrounding* preko svojstava *presents* (predstavlja), *describedBy* (opisano od strane) i *supports* (podržava), što se grafički može predstaviti kao na priloženoj slici.

1) Servisni profil

Servisni profil obezbjeđuje opis servisa i njegovog provajdera, i koristi se od strane registara servisa, kako bi se zatražio i promovisao servis. Ovo uključuje: opis servisa čitljiv ljudima, specifikaciju funkcionalnosti i funkcionalne atribute.

Veza prema klasi *Service* se opisuje sa dva svojstva: *presents* (predstavlja) i *presentedBy* (predstavljeno od). OWL-S definiše podklasu *Profile* klase *ServiceProfile*, kao moguće predstavljanje profila servisa. Ovo predstavljanje uključuje informacije čitljive ljudima, na primjer, naziv servisa, kratak opis servisa, te informacije o provajderu servisa, operateru za održavanje ili drugim osobama i organizacijama koje su odgovorne za servis.

Opis funkcionalnosti profila bavi se pitanjima transformacije informacija, predstavljanja ulaza i izlaza servisa, te ponašanjem i promjenom stanja njegovim izvršavanjem, što se izražava preduslovima i efektima. Ulazi i izlazi, u osnovi, semantički specificiraju vrste parametara sa kojima radi servis, odnosno koje se informacije zahtijevaju i koje se informacije produkuju od strane servisa. Preduslovi su logički izrazi koji određuju uslove koji se traže da bi servis mogao biti uspješno

izvršen. Efekti, koji se nazivaju i rezultatima, specificiraju izmjenjene koje servis izvrši svojim uspješnim izvođenjem.

Profili također uključuju svojstva koja se koriste za opisivanje osobina servisa. Ovo uključuje garancije kvaliteta koje su obezbjeđene od strane servisa, moguću kategorizaciju servisa, te druge informacije koje provajder servisa želi da podijeli. Servisi mogu imati različite prioritete, neki mogu biti brži, ali manje pouzdani, dok su drugi pouzdani i sigurni, ali zbog toga mogu biti sporiji. Ovo predstavlja kvalitet usluge, odnosno servisa, i ostaje na podnosiocu zahtjeva za servisom da provjeri kakav kvalitet se garantuje od strane pronađenog servisa.

2) Model servisa

Svaka *Service* instanca je opisana od strane bar jednog modela servisa. Klasa *ServiceModel* informiše klijente o tome kako koristiti servis. To se radi specificiranjem semantičkog sadržaja zahtjeva, odgovora, uslova pod kojima se određeni rezultati zadržavaju, te načinom na koji klijenti moraju prozvati servis.

Da bi se klijentu opisalo kako vršiti interakciju sa servisom, OWL-S posmatra servise kao *proces* sa skupom ulaza, izlaza, preuslova i efekata. Klasa *Process* je podklasa klase *ServiceModel* koja definiše model procesa. Model opisuje funkcionalna svojstva servisa, zajedno sa detaljima svojih konstitutivnih procesa.

Procesi mogu biti atomski, jednostavni i kompozitni. Atomski proces je opis servisa koji očekuje jednu dolaznu poruku i nakon toga daje povratnu poruku, odnosno izvršava se u jednom koraku. Poruke mogu biti kompleksne i sadržavati više ulaza i izlaza.

Jednostavni proces ne može biti direktno pozvan, ali slično kao i atomski proces, predstavlja proces koji se izvršava u jednom koraku. Jednostavni procesi se koriste kao apstraktni elementi. Mogu biti korišćeni za predstavljanje atomskih procesa ili da pruže pojednostavljen prikaz kompozitnih procesa.

Kompozitni proces zahtijeva nekoliko koraka u interakciji i/ili više serverskih akcija. Moguće je vršiti dekompoziciju ovih procesa u jednostavnije procese.

3) Pristupanje servisu

Klasa *ServiceGrounding* opisuje kako pristupiti OWL-S servisu. Ova klasa opisuje korištene protokole, format poruka za razmjenu, načine transporta i adresiranja. Profil i model servisa opisuju servis na višem, apstraktnom nivou, dok klasa za pristupanje servisu daje detalje konkretne implementacije.

Uopšteno govoreći, *ServiceGrounding* je mapiranje između parametara kojima rukuju atomski procesi i poruka koje nose ove parametre u određenom formatu pogodnom za prenos. OWL-S ne predefiniše jezik za opisivanje, ali zbog svoje široke upotrebe, obično se koristi jezik WSDL.

B. WSMO

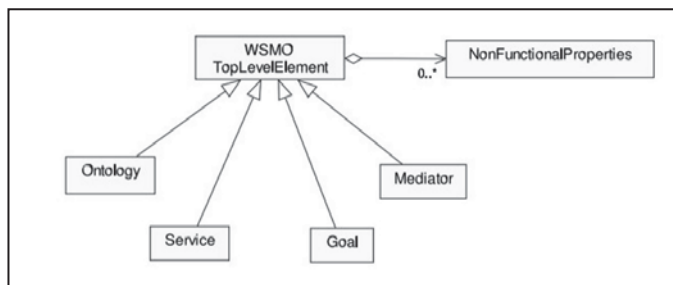
WSMO (*Web Service Modeling Ontology*) je konceptualni model za opisivanje različitih aspekata vezanih za semantičke

Web servise [2]. Njegov cilj je da razriješi problem integracije aplikacija na polju web servisa.

WSMO opisuje sve relevantne aspekte vezane za servise koji su dostupni kroz interfejs web servisa, kako bi se omogućila potpuna ili djelimična automatizacija zadataka, kao što su otkrivanje, izbor, kompozicija, medijacija, izvršavanje, monitoring i sl.

WSMO identifikuje četiri elementa najvišeg nivoa kao glavne koncepte koji moraju biti opisani kako bi se definisali semantički web servisi. Elementi su predstavljeni na Slici 2.

WSMO element Ontologija (*Ontology*) bavi se pitanjima uvođenja standardne terminologije za predstavljanje karakteristika servisa. Svi opisi resursa kao i svi podaci razmijenjeni za vrijeme izvršenja servisa su bazirani na ontologijama. One obezbjeđuju koncepte, relacije između ovih konceptata, te skup logičkih izraza (aksioma), koji istražuju semantiku konceptata i relacija. Pored toga, uvode se i pojmovi super-konceptata i super-relacija, koji predstavljaju mogućnost nasljeđivanja za koncepte i relacije.



Slika 2. WSMO elementi najvišeg nivoa

Servis (*Service*) obezbjeđuje konceptualni model za opisivanje svih aspekata servisa, uključujući njegova nefunkcionalna svojstva, kao i funkcionalnost i interfejs. Tako nedvosmislen model servisa sa dobro definisanom semantikom može biti procesiran i interpretiran od strane računara, bez intervencije čovjeka, omogućujući automatizaciju zadataka vezanih za upotrebu web servisa, kao što su otkrivanje, izbor, kompozicija, medijacija, izvršenje ili monitoring.

Ciljevi (*Goals*) se u WSMO koriste za opisivanje korisničkih zahtjeva za servisom. Oni predstavljaju reprezentacije ciljeva koji trebaju biti ispunjeni kroz izvršavanje servisa. Pored toga, oni mogu biti opisi servisa koji bi potencijalno zadovoljili želje korisnika. Za definisanje korištene terminologije za opis relevantnih aspekata ciljeva koriste se ontologije.

Medijator (*Mediator*) opisuje aspekte koji se brinu o problemima međudjelovanja između različitih WSMO elemenata. Medijatori su bitan koncept za razriješenje nekompatibilnosti podataka, procesa i nivoa protokola. WSMO definiše četiri tipa medijatora: *OO Mediator*, *GG Mediator*, *WG Mediator* i *WW Mediator*. Svi medijatori su podklase glavne klase *Mediator*. Prefiks u njihovim nazivima označava komponente koje se povezuju.

OO medijatori se brinu o heterogenosti između ontologija. Izvorne komponente su ontologije ili drugi OO medijatori koji

su heterogeni i trebaju biti integrirani. GG medijatori povezuju ciljeve, omogućujući kreiranje novog cilja, odnosno ciljnih ontologija. WG medijatori povezuju web servise sa ciljem, i razriješavaju terminološka neslaganja i pronalaze funkcionalne razlike ako postoje. WW medijatori se koriste za uspostavljanje međudjelovanja između web servisa. Ovaj medijator vrši medijaciju između koreografija web servisa koji trebaju međudjelovati, i zbog kojih može biti potrebna medijacija na nivou podataka, protokola ili procesa.

C. METEOR-S

Projekat METEOR-S (*Managing End-To-End Operations-Semantics*) fokusira se na korištenje semantike kroz kompletan životni ciklus semantičkih web procesa. Procesi reprezentuju složene interakcije među semantičkim web servisima. METEOR-S cilja na određena bitna područja životnog ciklusa semantičkih web procesa, kao što su označavanje, objavljivanje, otkrivanje, povezivanje, kompoziciju, te izvršavanje web servisa. Ono po čemu se METEOR-S razlikuje od ranije opisanih inicijativa semantičkih web servisa je snažno držanje za postojeće standarde web servisa. Ustvari, filozofija ovog pristupa je da se postojeći standardi inkrementalno proširuju semantikom.

1) Semantičko označavanje

U središtu inicijative METEOR-S leži MWSAF (*METEOR-S Web Service Annotation Framework*) koji predstavlja okvir za (polu)automatsko semantičko označavanje, odnosno opisivanje web servisa [3]. Ovo označavanje se odnosi na četiri različita aspekta semantike web servisa:

- MWSAF podržava uključivanje oznaka o semantici ulaza i izlaza web servisa,
- podržava definisanje funkcionalne semantike (na primjer, šta servis radi),
- omogućuje uključenje izvršne semantike kako bi se podržala verifikacija korektnosti izvršavanja web servisa,
- uključuje informacije vezane za kvalitet servisa (QoS), kao što su performanse ili cijena koštanja izvršenja web servisa.

Prvobitna istraživanja u razvoju ovog okvira bila su posvećena podržavanju (polu)automatskog označavanja *XML Schema* dijela definicije web servisa. Ovo je bazirano na transformaciji *XML Schema* i ontologija u jedinstven format reprezentacije pod nazivom *SchemaGraph*, kako bi se omogućila usklađenost među modelima. Nakon transformacije skup algoritama za usklađivanje može biti primjenjen (polu)automatski kako bi se obogatile sintaksne definicije semantičkim oznakama.

MWSAF se sastoji iz tri glavne komponente: prostor za ontologije, biblioteka metoda usklađivanja (*matchmaking*) i biblioteka transformacija. Prva komponenta čuva ontologije koje će biti korištene za opisivanje web servisa. Biblioteka metoda usklađivanja obezbjeđuje različite algoritme implementacije za lingvističko i strukturalno usklađivanje između koncepata i elemenata web servisa. Konačno, biblioteka transformacija sastoji se iz programa koji se koriste

za generisanje *SchemaGraph* reprezentacije ontologija i web servisa.

MWSAF pruža podršku u opisivanju web servisa pretraživanjem i računanjem saglasnosti između domenskih modela i elemenata web servisa. Zadnji korak u procesu opisivanja je reprezentacija za buduću ponovnu upotrebu, što se omogućuje tehnologijom WSDL-S.

WSDL-S predstavlja pristup za pridruživanje semantičkih oznaka web servisima korištenjem postojećih standarda web servisa. Konkretno, semantičke oznake, u formi URI referenci prema vanjskim modelima, dodaju se interfejsima, operacijama i porukama u WSDL jeziku. WSDL-S ne postavlja bilo kakva ograničenja prema izboru jezika za reprezentaciju znanja koji se koristi za definisanje semantičkih modela. Mogu se koristiti WSML, OWL, UML i drugi jezici.

SAWSDL (*Semantic Annotated WSDL*) je novija, restriktivnija i homogenizovanija verzija WSDL-S u kojoj se neke oznake, kao na primjer preduslovi i efekti, eksplicitno ne razmatraju, obzirom da za njih, u krugovima istraživača semantičkih web servisa, trenutno ne postoji dogovor kako ih modelovati. Ipak, SAWSDL ne isključuje potpuno uvrštavanje ovih oznaka. Jedna od prednosti u odnosu na WSDL-S je činjenica da SAWSDL dozvoljava uključivanje struktura semantičkog znanja iz vanjskih ontologija.

2) Otkrivanje i objavljivanje servisa

MWSDI (*METEOR-S Web Services Discovery Infrastructure*) je skalabilna infrastruktura za objavljivanje i otkrivanje web servisa bazirana na semantici. MWSDI omogućuje jedinstven pristup velikom broju nezavisnih registara i implementirana je korištenjem P2P (*peer-to-peer*) tehnike, kako bi se omogućila skalabilnost i fleksibilnost. Ova infrastruktura koristi semantiku za dvije namjene. Prvo, koristi tzv. registarsku ontologiju, koja smješta informacije o registrima, održava veze među domenima unutar MWSDI, te im pridružuje registre. Ova ontologija čuva mapiranje između registara i domena, tako da pronalaženje web servisa za određen domen može biti usmjereno na odgovarajući registar. Drugo, MWSDI predviđa uključenje domenski specifičnih ontologija za registre, tako da web servisi mogu biti opisani mapiranjem ulaza i izlaza na postojeće domenske ontologije. Svrha definisanja ovog mapiranja je omogućavanje semantičkog otkrivanja dozvoljavajući korisnicima da izraze svoje zahtjeve kao servisne šablone, koji su izraženi korištenjem koncepata iz iste ontologije.

Semantičko objavljivanje servisa u MWSDI registrima koristi UDDI *tModels* za registrovanje domenskih ontologija i *CategoryBags* za kategorisanje WSDL entiteta prema jednom ili više *tModels*-a. MWSDI obezbjeđuje ručni i poluautomatski mehanizam za definisanje mapiranja između WSDL elemenata i koncepata u domenskim ontologijama.

3) Medijacija

METEOR-S se bavi problemom heterogenosti među razmijenjenim porukama tako što se ulazi i izlazi servisa mapiraju na konceptualne modele. Ova mapiranja su uglavnom podržana WSDL-S i SAWSDL opisima. WSDL-S obezbjeđuje *shemaMapping* anotacije, dok SAWSDL vrši razdvajanje u *liftingSchemaMapping* i *loweringSchemaMapping*, gdje je

podizanje (*lifting*) proces koji transformiše sintaksne informacije u semantičke podatke, a spuštanje (*lowering*) obrnuti proces.

4) Kompozicija

Kompozicija web servisa je podržana preko MWSCF (*METEOR-S Web Service Composition Framework*). U suštini, koriste se semantički šabloni procesa koji procese posmatraju kao skup semantički predstavljenih aktivnosti. Bazirano na ovakvim šablonima, izvršni procesi mogu biti generisani povezivanjem semantički definisanih aktivnosti za konkretne web servise koji se prilagođavaju specifikacijama aktivnosti. MWSCF se sastoji iz četiri komponente: kreator procesa, infrastruktura za otkrivanje, XML repozitorijumi i sistem za izvršenje procesa.

III. POREĐENJE

U ovom dijelu rada biće upoređene inicijative semantičkih web servisa: OWL-S, WSMO i METEOR-S, te će se identifikovati sličnosti i razlike u pristupima.

OWL-S i WSMO koriste pristup "odozgo prema dole", gdje se obezbjeđuju ontologije visokog nivoa koje pružaju ekspresivne okvire za opisivanje različitih koncepata web servisa, što ima za cilj automatizovano otkrivanje, pozivanje, kompoziciju i praćenje web servisa. Pri tome se standardi web servisa koriste samo za pitanja pristupanja i pozivanja. S druge strane, METEOR-S je zasnovan na postojećim standardima web servisa, koje inkrementalno proširuje podrškom za semantiku, tako što im se pridružuju specifične ekstenzije koje povezuju sintaksne definicije sa njihovim semantičkim oznakama.

Po pitanju usaglašenosti sa standardima web-a, sve tri inicijative su bazirane na principu identifikovanja resursa korištenjem web identifikatora, odnosno URI (*Unified Resource Identifiers*). Takođe, podržavaju koncept imenskih prostora (*namespaces*) za konzistentno označavanje prostora informacija, kao i XML jezik i ostale W3C web standarde. To nam omogućuje integraciju postojećih web resursa i primjenu probanih i testiranih alata i biblioteka.

Kod sve tri inicijative, svi opisi resursa, kao i svi podaci koji se razmjene za vrijeme korišćenja servisa, bazirani su na ontologijama. Korišćenje ontologija nam omogućuje napredno procesiranje informacija i automatizovano rješavanje problema heterogenosti među resursima.

Kod OWL-S pristupa koriste se ontologije izgrađene u OWL jeziku. WSMO ontologije su opisane na meta nivou korištenjem MOF modela. Ovdje su model i jezici razdvojeni jedni od drugih. Ovaj mehanizam nudi veću fleksibilnost, bez diktiranja formalizma (na primjer, specifičnog ontološkog jezika) u opisivanju uvezenih ontologija. Kod METEOR-S programeri web servisa mogu slobodno označavati web servise ontološkim jezikom po ličnom izboru (na primjer, UML, RDF-S, OWL). Ovo je značajno pošto sposobnost ponovnog korištenja postojećih domenskih modela, izražena u jezicima za modelovanje, može značajno ublažiti potrebu za odvojenim modelovanjem semantike. [4]

Resursi, kao što su ontologije, opisi web servisa, ciljevi i medijatori, trebaju biti striktno razdvojeni jedni od drugih. To znači da bi resursi trebalo da se razvijaju izolovano, bez obzira na njihovu moguću upotrebu ili interakciju sa drugim resursima. Sva tri pristupa omogućuju da se ontologije i opisi servisa definišu nezavisno.

Razdvojenost resursa dovodi do pojave heterogenosti među njima, tako da je neophodno uspostaviti mehanizam medijacije koji će se brinuti o različitostima podataka, ontologija, protokola i procesa. Medijatori bi trebalo da budu elementi najvišeg nivoa u tehnologijama semantičkih web servisa.

U WSMO pristupu definišu se četiri tipa medijatora: *OO Mediator*, *GG Mediator*, *WW Mediator* i *WG Mediator*, zavisno od toga da li se brinu o heterogenosti između ontologija, ciljeva, web servisa ili između ciljeva i web servisa. OWL-S eksplicitno ne uzima u obzir problem heterogenosti u samom jeziku, ostavljajući to kao arhitekuralno pitanje. Tako medijatori nisu dio ontologije, već dio infrastrukture web servisa. METEOR-S takođe ne nudi direktnu podršku po pitanju medijacije. Moguće je vršiti medijaciju podataka kroz mapiranje ulaza i izlaza preko WSDL-S i SAWSDL oznaka.

Konteksti unutar kojih funkcionišu klijenti, odnosno podnosioci zahtjeva za servisom, kao i provajderi servisa mogu biti veoma različiti, tako da bi okvir za razvoj semantičkih web servisa trebao razlikovati zahtjeve za servisom i opise funkcionalnosti koje provajderi servisa mogu ponuditi.

OWL-S servisni profil predstavlja predviđenu namjenu servisa, i opisuje kako funkcionalnost ponuđenu od strane provajdera, tako i funkcionalnost traženu od podnosioca zahtjeva. WSMO razlikuje tačke gledišta provajdera i podnosioca zahtjeva, uvodeći dva različita koncepta: cilj i sposobnost. Konkretno, cilj specificira šta klijent želi kada konsultuje web servis, dok sposobnost definiše servis obzirom na to šta nudi podnosiocu zahtjeva, na primjer, koju funkcionalnost servis pruža. S druge strane, analogno OWL-S servisnom profilu i WSMO cilju i sposobnosti je apstraktni dio WSDL 2.0 ekstenzije okvira METEOR-S, pod nazivom WSDL-S, u okviru kojeg se ne razdvajaju uloge provajdera i klijenta. [4]

Nefunkcionalna svojstva u servisnom profilu OWL-S (kao što su naziv servisa, opisi čitljivi ljudima i kontakt informacije) nisu eksplicitno bazirana na standardnim specifikacijama meta podataka. WSMO preporučuje upotrebu široko prihvaćenih rječnika, kao što su *Dublin Core* i *Friend-of-a-Friend*. Takođe, nefunkcionalna svojstva mogu biti izražena u bilo kom elementu WSMO, dok je kod OWL-S to moguće samo u servisnom profilu. [5]

OWL-S obezbjeđuje detaljan opis Modela servisa, omogućujući definisanje atomskih, jednostavnih i složenih procesa, zajedno sa tokom upravljačkih informacija i podataka.. WSMO klasa *Service* odgovara Modelu servisa kod OWL-S, i ona opisuje funkcionalnost servisa zajedno sa interfejsima. Tako se WSMO sposobnost može posmatrati analogno konceptu procesa kod OWL-S, preciznije atomskom procesu, dok se jednostavni procesi ne uzimaju u obzir kod WSMO. Kod METEOR-S pristupa, operacija web servisa je

opisana konkretnim dijelom WSDL-S dokumenta i modelom apstraktnog procesa BPEL4WS.

Kad se transformacija informacije i promjena stanja uzimaju u obzir, tri inicijative uvode veoma slične koncepte. Ulazi i izlazi OWL-S procesa, kao i preduslovi i postuslovi WSMO sposobnosti, zajedno sa semantičkim označavanjem ulaza i izlaza METEOR-S okvira se opisuju transformacijom informacije. Pored toga, za opis promjene stanja OWL-S uvodi preduslove i rezultate procesa, WSMO pretpostavke i efekte sposobnosti, a METEOR-S preduslove i efekte.

OWL-S ne pravi jasnu razliku između koreografije i orkestracije servisa. Ovdje nedostaje formalni model, iako neki spoljni radovi definišu formalnu semantiku OWL-S procesa. OWL-S definiše samo jedan modela servisa, tako da postoji samo jedan način za interakciju sa servisom. WSMO specificira koreografiju i orkestraciju u interfejsu opisa web servisa. Koreografija opisuje spolja vidljivo ponašanje servisa, dok orkestracija opisuje šta drugi servisi moraju da urade ili kakvi drugi ciljevi moraju biti ispunjeni da bi se obezbijedila funkcionalnost servisa. WSMO dozvoljava definisanje višestrukih interfejsa za pojedinačan servis, pošto se očekuje da može biti više načina interakcije sa servisom. [5]

Po pitanju pristupanja web servisu, sve tri inicijative se baziraju na mapiranju u WSDL jezik. OWL-S je ovdje u prednosti pošto je kod njega ovaj koncept podignut na najviši nivo modela.

Da bi se omogućila podrška za neke drugačije ili novije standarde, neophodno je da postoji razdvojenost između opisa servisa i načina njegove implementacije, odnosno između opisa elemenata semantičkih web servisa i tehnologija na kojima se oni izvršavaju.

OWL-S se bazira na odgovarajućim formalizmima koji mu omogućuju jasne semantičke opise i otvoren je za različite implementacione tehnike. WSMO obezbjeđuje ontološki opisni model koji može biti u saglasnosti kako sa postojećim, tako i sa budućim tehnologijama. METEOR-S je snažno uparen sa postojećim standardima web servisa.

IV. ZAKLJUČAK

Zadnjih nekoliko godina semantički web servisi dobijaju sve više pažnje. Postoji nekoliko paralelnih inicijativa na područjima istraživanja i razvoja tehnologija semantičkih web servisa, ali, iako imaju iste ciljeve, one pokazuju znatne razlike u pristupima.

Nijedna od opisanih inicijativa još uvijek ne nudi kompletno rješenje, ali svaka od njih ima određene komplementarne prednosti u odnosu na druge. Tako OWL-S obezbjeđuje bogatu XML baziranu servisnu ontologiju, jasno definisan procesni model i detaljne specifikacije koreografije i pristupanja web servisu. Za razliku od ostalih inicijativa, WSMO razdvaja tačke gledišta klijenta i provajdera. WSMO obezbjeđuje formalnu semantiku za koreografiju web servisa, pruža bolju slojevitost jezika i korišteni jezici prevazilaze problem kombinovanja OWL sa različitim jezicima za kreiranje pravila [6]. METEOR-S se snažno pridržava postojećih standarda web servisa, inkrementalno ih proširujući

semantikom. Tako jezici na kojima je METEOR-S zasnovan, WSDL-S i SAWDSL, omogućuju jednostavno dodavanje semantike u postojeće web servise.

WSMO pristup opisivanju korisničkih zahtjeva možemo posmatrati prirodnijim, u odnosu na OWL-S opis zahtjeva u formi željenog profila. Takođe, značajno je uključivanje medijatora kao modelirajućeg elementa u WSMO pristupu, za razliku od OWL-S i METEOR-S gdje se to prepušta infrastrukturi web servisa.

WSMO djeluje kao konceptualno jači pristup, ali tek treba defisati neke otvorene aspekte kako bi postao iskoristiv u realnim situacijama.

Možemo zaključiti da su tehnologije semantičkih web servisa još uvijek daleko od finalnog proizvoda. Moguće je kreirati distribuisane aplikacije koje se zasnivaju na izvršavanju web servisa na WWW infrastrukturi, međutim, one i dalje zahtijevaju prisustvo čovjeka u određenim trenucima, kao na primjer pri izboru servisa raspoloživih u registrima. Tehnologije semantičkog weba mogu biti iskorištene za opisivanje i rezonovanje mogućnosti web servisa [7]. Ipak, i dalje ostaju neriješena neka značajna pitanja, od kompozicije servise, pa do povjerenja servisa i sigurnosnih aspekata, što će biti predmet istraživanja u narednim godinama.

LITERATURA

- [1] K. Breitman, M.A. Casanova, W. Truszkowski, "Semantic web – Concepts, Technologies and Applications", Springer, 2007.
- [2] C. Feier, "WSMO Primer, WSMO Deliverable D3.1", WSMO Working Draft, 2005.
- [3] D. Fensel, F.M. Facca, E. Simperl, I. Toma, "Semantic Web Services", Springer, 2011.
- [4] P. Bouros, "Semantic Web Services: A conceptual comparison of OWL-S, WSMO and METEOR-S approaches", National and Kapodistrian University of Athens, 2006.
- [5] J. de Bruijn, D. Fensel, M. Kifer, J. Kopeck, R. Lara, H. Lausen, "Relationship of WSMO to other relevant technologies", W3C Member Submission, 2005.
- [6] R. Lara, A. Polleres, H. Lausen, D. Roman, J. de Bruijn, D. Fensel, "A Conceptual Comparison Between WSMO and OWL-S", DERI, 2005.
- [7] L. Cabral, J. Domingue, E. Motta, T. Payne, F. Hakimpour "Approaches to Semantic Web Services: An Overview and Comparisons", Knowledge Media Institute, The Open University, Milton Keynes, 2004.

ABSTRACT

Web services technologies mostly consider syntactic aspects of services and their adaptation to dynamic environments, such as modern WWW, is almost impossible without human assistance. The initiatives for semantic annotation of web services have appeared in an effort to enhance interaction between classic web services, so they can be functional in environments with heterogeneous data and different behaviour of resources. Three initiatives of semantic web services are presented in this work. We analyse their different concepts, estimate their current capabilities and review their development directions.

SEMANTIC WEB SERVICES – THE INITIATIVES FOR SEMANTIC ANNOTATION OF WEB SERVICES

Velibor Radulović, Čedomir Radulović, Zoran Đurić