

hyväksymispäivä arvosana

arvostelija

WSMX - Web Service Execution Environment

Ville Raustia

Helsinki 2.4.2009

Seminaariraportti

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Ville Raustia			
Työn nimi — Arbetets titel — Title			
WSMX - Web Service Execution Environment			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Seminaariraportti		2.4.2009	13 sivua
Tiivistelmä — Referat — Abstract			
<p>Web Service Modeling eXecution environment (WSMX) on semanttisiin web-palveluihin keskittyvä suoritusalue. Se tarjoaa suoritustympäristön semanttisten web-palveluiden löytämiseen, valitsemiseen, koostamiseen ja kutsumiseen.</p> <p>WSMX on avoimen lähdekoodin toteutus, joka perustuu semanttisten web-palveluiden mallinnusontologiaan (WSMO, Web Service Modeling Ontology). WSMO tarjoaa käsitteelliset mallit, joiden pohjalta WSMX on toteutettu.</p>			
Avainsanat — Nyckelord — Keywords			
WSMX, WSMO, B2B middleware, SOA, Semantic Web Services			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — övriga uppgifter — Additional information			

Sisältö

1	Johdanto	1
2	Tekniikoiden rajoitteita	2
3	WSMO - Web Service Modeling Ontology	3
3.1	Ontologiat	3
3.2	Web-palvelut	4
3.3	Maalit	6
3.4	Välittäjät	6
4	WSMX - Web Service Execution Environment	8
4.1	Komponentit	8
4.2	Web-palveluiden etsiminen	9
4.3	Ulkoinen toiminta	10
4.3.1	achieveGoal	10
4.3.2	discoverWebServices ja invokeWebService	11
4.4	WSMX P2P-verkossa	11
5	Yhteenveto	12
	Lähteet	13

1 Johdanto

W3C (The World Wide Web Consortium) määrittelee nykyisellään web-palvelun olevan ohjelmistojärjestelmä, joka mahdollistaa yhteensopivien tietokoneiden kanssakäymisen verkon välityksellä tarkasti määriteltyjen, koneellisesti luettavien rajapintojen kautta [W3C04]. Muut järjestelmät kommunikoivat web-palvelun kanssa lähinnä SOAP-viestien välityksellä. Erilaisten web-palveluiden kirjo on suuri ja toteutustavat usein erilaisia ja eri tahoille räätälöityjä. Vaikka kanssakäynti tapahtuukin tietokoneiden välityksellä, vaativat web-palvelut nykyisellään oraakkelia (ts. ihmistä) etsimään ja valitsemaan palvelun, joka ottaa vastaan oikeanlaiset syötteen ja tuottaa niistä mieluisan tuloksen.

Jotta web-palveluiden etsiminen, valinta, koostaminen ja suoritus voitaisiin kokonaan automatisoida, täytyy niille lisätä semanttiset kuvaukset. Siinä missä tavantomaiset web-palvelut toimivat yhdessä lähinnä syntaktisten ominaisuuksiensa puolesta, semanttiset web-palvelut (Semantic Web Services, SWS) nostavat palveluiden merkityksen korkeammalle tasolle semanttisten kuvausten avulla.

WSMX (Web Service Modeling Execution Environment) on väliohjelmisto, joka automatisoi SWS:n etsimisen, valinnan, koostamisen ja suorituksen. Yksinkertaisimmillaan käyttäjä voi määrittellä käytettävän ontologian ja halutun tavoitteen, jolloin WSMX pystyy etsimään oikeanlaisen web-palvelun, kutsua sitä ja lopuksi palauttaa palvelun tuottaman tuloksen käyttäjälle sopivassa formaatissa.

Tämän raportin tarkoituksena on antaa lukijalle yleiskuva WSMX:stä ja operaatioista jotka sillä pystytään automatisoimaan. Lisäksi tutustutaan WSMO:hon (Web Service Modeling Ontology), joka tarjoaa käsitteelliset mallit SWS:ien toteuttamiseen.

2 Tekniikoiden rajoitteita

Kuten aikaisemmin todettiin, on nykyisissä web-palveluissa ja niihin liittyvissä tekniikoissa puutteita muun muassa yhteentoimivuuden ja etsimisen suhteen. Nämä ongelmat tulevat esille varsinkin B2B-ympäristöissä, joissa palveluiden integrointi ympäristöistä toisiin voi tuottaa paljonkin ylimääräistä työtä [VMK⁺07].

WSDL, UDDI ja SOAP tarjoavat ratkaisuja web-palveluiden kuvaamiseen, etsimiseen ja viestinvälitykseen. Niissäkin on kuitenkin puutteita semanttisten web-palveluiden kannalta. UDDI:sta voi listata ja etsiä web-palveluita. Palvelut on kuitenkin kuvattu ainoastaan ihmisymmärrettävässä muodossa, joka mahdollistaa vain hakusanoihin perustuvan palveluhaun.

Palveluiden ja niiden käyttäjien välillä kulkeva tieto voi aiheuttaa ongelmia niin tiedon muodon kuin sen merkityksenkin kannalta. Tiedon merkityksen voi toki sopia toimijoiden kesken, mutta automatisoinnin kannalta merkityksen joutuu selvittämään itse. Tiedonvaihtoon liittyvät semanttiset kuvaukset tuovat ratkaisun ongelmaan ja selkeyttävät pyyntö-vastaus -tyylistä kommunikointia monimutkaisempia-kin kuvioita.

3 WSMO - Web Service Modeling Ontology

Pelkät käsitteelliset kuvaukset eivät riitä määrittelemään semanttisia web-palveluita. Niiden lisäksi SWS:lle pitää määrittellä erikseen semanttiset ominaisuudet ontologiamallin avulla. WSMO on WSMF-kehukseen (Web Service Modeling Framework) perustuva ontologia semanttisten web-palveluiden eri ominaisuuksien kuvaamiseen [RLK06]. Se on myös malli, johon koko WSMX:n toteutus perustuu.

WSMO:n kuvaamat käsitteet voidaan mallintaa WSML-kielellä (Web Service Modeling Language). WSML on WSMO:n pohjalta muodostettu formaali kieli, jolla on mahdollista kuvata kaikki WSMO:n määrittelemät käsitteet [dBLK⁺05]. WSML:n eri variaatiot ovat WSML-Core, DSML-DL, WSML-Flight, WSML-Rule ja WSML-Full, joista core on ilmaisukyvyltään heikoin, mutta laskennallisesti tehokkain ja full ilmaisukyvyltään voimakkain, mutta laskennallisesti vaativin.

WSMO on suunniteltu muun muassa seuraavia periaatteita käyttäen:

- **Yhteensopivuus webin kanssa**
Resurssit tunnistetaan niiden URI:n perusteella ja niitä voidaan niputtaa yhteen nimiavaruuksilla (namespace)
- **Ontologiaperusteisuus**
Kaikki tietomallit perustuvat ontologioihin. Ontologiaperusteisuus mahdollistaa semanttisesti tehostetun tiedonkäsittelyn.
- **Resurssien sitoutumattomuus (strict decoupling)**
Resurssit ovat itsenäisiä, muiden resurssien muutoksista riippumattomia elementtejä. Sitoutumattomuus mahdollistaa komponenttien hajauttamisen sekä helpottaa niiden muokkaamista ja vaihtamista.

WSMO:n tärkeimmät komponentit ovat ontologiat, maalit, web-palvelut ja välittäjät (mediators), joista kerrotaan seuraavissa kappaleissa enemmän.

3.1 Ontologiat

Ontologia (ks. kuva 1) on formaali eksplisiittinen spesifikaatio jaetusta käsitteistöstä [FD05]. WSMO:ssa ne ovat avainasemassa, sillä niiden avulla kuvataan kaikki elementit. Ontologioilla kuvataan myös resurssien formaali semantiikka, niiden relaatiot toisiin resurssihin ja yhdistetään puhekielen ja konekielen terminologiat

toisiinsa. Ontologioiden muodostamisen helpottamiseksi niitä voi myös sisällyttää toisiinsa. Sisällyttämisen aiheuttamat konfliktit voidaan tarvittaessa selvittää ontologiavälittäjällä (ooMediator). Ontologian peruselementteihin WSMO:ssa kuuluvat muun muassa:

- **Käsitteet**

Määrittelevät jonkin toimialueen yhteisesti hyväksytyn terminologian. Käsitteen määrittelee sen oma hierarkia ja attribuutit, jotka voivat olla joko tietotyyppejä tai toisia käsitteitä. Käsite voi myös olla yhden tai useamman käsitteen alakäsite, jolloin se perii kaikki yläkäsitteen ominaisuudet ja rajoitteet. Lisäksi käsitteen instanssi on samalla sen yläkäsitteen instanssi.

- **Relaatiot**

Määrittelevät vuorovaikutussuhteet käsitteiden tai käsitteiden instanssien välillä. Myös relaatiolla voi olla ylärelaatioita, jolloin se perii ylärelaation ominaisuudet ja rajoitteet.

- **Instanssit**

Käsitteiden tai relaatioiden ilmentymiä. Instansseilla on konkreettiset arvot attribuuteille ja parametreille.

- **Aksioomat**

Logiikkalausekkeita, joilla avustetaan formalisoimaan toimialueelle ominaista tietoa.

3.2 Web-palvelut

Web-palvelu on resurssi, jota kutsumalla pystytään saavuttamaan tavoitettava maali. Web-palvelun koreografia määrittelee palvelun toiminnan sen käytön aikana. Käyttäjän on noudatettava palvelun koreografiaa, mikäli haluaa käyttää sitä.

Web-palvelun capability-elementti kertoo sen toiminnallisuuden alku- ja jälkiehtojen, oletuksien ja palvelun vaikutuksien avulla. Yhdellä web-palvelulla on ainoastaan yksi capability-elementti.

Interface-elementti kertoo lisätietoa siitä, kuinka web-palvelun toiminnallisuus saavutetaan. Se sisältää choreography-elementin, joka kertoo palvelun kommunikointitavan ja orchestration-elementin, jolla ilmaistaan kuinka palvelun toiminnallisuus saavutetaan yhteistyössä muiden palveluntarjoajien kanssa. WSMO:ssa koreografia

```

namespace { _ "http://example.org/BookTicketInterfaceOntology#",
  dc _ "http://purl.org/dc/elements/1.1#",
  tr _ "http://example.org/tripReservationOntology#",
  wsm1 _ "http://www.wsmo.org/wsm1/wsm1-syntax#",
  po _ "http://example.org/purchaseOntology#"
}

ontology _ "http://example.org/BookTicketInterfaceOntology#"
  nonFunctionalProperties
    dc#title hasValue "Book Ticket Interface Ontology"
    dc#creator hasValue "DERI Innsbruck"
    dc#description hasValue "an ontology that redefines concepts and
      relations from other ontologies in order to reuse them in the
      choreography and orchestration; two additional non-functional
      properties are defined for the targeted concepts and relations:
      mode and grounding"
    dc#publisher hasValue "DERI International"
  endNonFunctionalProperties

  importsOntology { _ "http://www.example.org/tripReservationOntology",
    _ "http://www.wsmo.org/ontologies/purchaseOntology"
  }

```

Kuva 1: Esimerkki ontologian määrittelystä (ilman käsitteitä)[FD05]

```

namespace { _ "http://example.org/bookTicket#",
  dc _ "http://purl.org/dc/elements/1.1#",
  tr _ "http://example.org/tripReservationOntology#",
  foaf _ "http://xmlns.com/foaf/0.1/",
  wsm1 _ "http://www.wsmo.org/wsm1/wsm1-syntax#",
  bti _ "http://www.example.org/BookTicketInterfaceOntology#"}

webService _ "http://example.org/bookTicketWebService"

importsOntology _ "http://example.org/tripReservationOntology"

capability BookTicketCapability
interface BookTicketInterface

```

Kuva 2: Esimerkki Web-palvelun määrittelystä [FD05]

```

Class capability sub-Class wsmoElement
  importsOntology type ontology
  usesMediator type {ooMediator, wgMediator}
  hasNonFunctionalProperties type nonFunctionalProperty
  hasSharedVariables type sharedVariables
  hasPrecondition type axiom
  hasAssumption type axiom
  hasPostcondition type axiom
  hasEffect type axiom

```

Kuva 3: Capability-elementin sisältö [RLK06]

siis määrittää kahden toimijan (web-palvelun ja sitä kutsuvan tahon) välisen kommunikointimallin ja orkestrointi palvelun ja useamman eri toimijan välisen yhteistyön, kun pyrkimys on saada aikaiseksi palvelun tuottama tulos [RLK06].

3.3 Maalit

Maali määrittelee tavoitteet, joita asiakas odottaa ja haluaa web-palvelun toteuttavan omasta näkökulmastaan. Maalia käytetään web-palvelun etsinnän aikana, jolloin sitä vertaillaan löydettyjen palveluiden tuottamiin tuloksiin.

```

namespace { "http://example.org/goals#",
  dc _ "http://purl.org/dc/elements/1.1#",
  tr _ "http://example.org/tripReservationOntology",
  wsm1 _ "http://www.wsmo.org/wsm1/wsm1-syntax#",
  loc _ "http://www.wsmo.org/ontologies/locationOntology#"}

goal _ "http://example.org/havingATicketReservationInnsbruckVenice"
imports Ontology { _ "http://example.org/tripReservationOntology",
  _ "http://www.wsmo.org/ontologies/locationOntology"}
capability
  postcondition
    definedBy
      ?reservation[
        reservationHolder hasValue ?reservationHolder,
        item hasValue ?ticket
      ] memberOf tr#reservation
    and
      ?ticket[
        trip hasValue ?trip
      ] memberOf tr#ticket
    and
      ?trip [
        origin hasValue loc#innsbruck,
        destination hasValue loc#venice
      ] memberOf tr#trip.

```

Kuva 4: Esimerkki maalin määrittelystä [FD05]

Capability-elementin sisältö määrittelee mitä web-palvelun pitäisi tehdä. Web-palvelussa palvelun toiminnallisuus määritettiin samalla elementillä. Esimerkissä on määritelty ainoastaan palvelun jälkiehdot, mikä tässä tapauksessa on se, että asiakkaalla on hallussaan lippu varatulle matkalle. Tarvittaessa asiakas voi määrittellä interface-elementissä myös tavan, jolla hän haluaa kommunikoida web-palvelun kanssa.

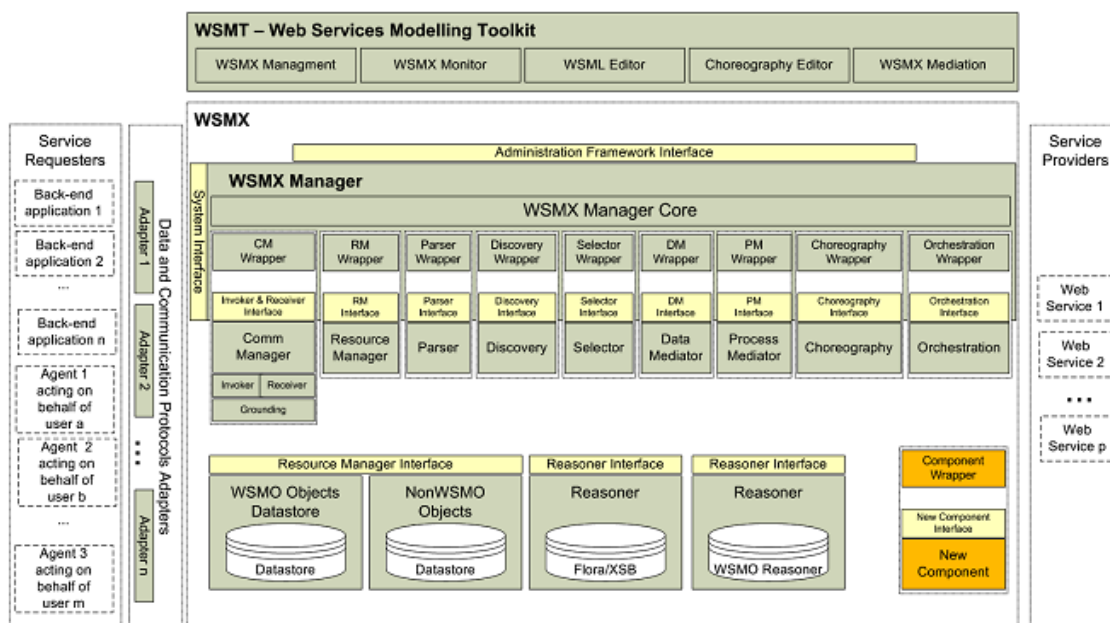
3.4 Välittäjät

Välittäjien avulla voidaan liittää erilaisia, tai lähinnä eri muodossa olevia resursseja toisiinsa [FD05, RLK06]. Välittäjät siis ratkovat erilaiset yhteensopivuuson-

gelmat resurssien välillä. WSMO:ssa on välittäjät maalien yhdistämiseksi (ggMediator), ontologioiden sisällyttämiseksi ja niiden vivahdeerojen korjaamiseksi (ooMediator), web-palveluiden muuntamiseksi maaleiksi (wgMediator) ja kahden web-palvelun yhdistämiseksi (wwMediator).

4 WSMX - Web Service Execution Environment

WSMX tarjoaa toiminnot semanttisten web-palveluiden löytämiseen, valintaan, välitykseen, suoritukseen ja yhteensovittamiseen [MZ05]. Se on avoimeen lähdekoodiin perustuva WSMO:n pohjalta toteutettu suoritusympäristö. Kaikki WSMX:ssä toteutetut toiminnot ja ominaisuudet tukevat WSMO:ssa esitellyjä käsitelmällejä.



Kuva 5: WSMX arkkitehtuuri [MZ05]

WSMX:n nykyversio on 0.5 ja se on ladattavissa WSMX:n kotisivuilta (<http://www.wsmx.org/>). WSMX:n javalla toteutettu ja sisäiset WSMO-objektit aiemmin esitetyllä WSML-kielellä.

4.1 Komponentit

WSMX:n sisäinen arkkitehtuuri perustuu palvelusuuntautuneeseen järjestelmään ja siinä painottuu vahvasti komponenttien itsenäisyys (strong decoupling). Komponentteja voidaan lisätä ja poistaa järjestelmästä myös ajon aikana. Jokaiselle komponentille on julkinen rajapinta, jonka kautta päästään käsiksi sen tarjoamiin palveluihin. Komponentteja voi siis vaihtaa ja kehittää itse, kunhan ne toteuttavat rajapintojen määrittelemät palvelut.

Core-komponentti on WSMX:n keskeisin osa - ja itse asiassa ainoa keskitetty komponentti koko järjestelmässä - joka hallitsee kaikkia muita komponentteja. Se käsittelee WSMX:n toimintalogiikan, tapahtumat, sisäiset tapahtumaketjut (workflow), hajautettujen komponenttien lataamiset ja niin edelleen. Core-komponentin keskityneisyys muuttuu mahdollisesti tulevaisuudessa, mikäli WSMX:n työryhmä kehittää sille tuen hajautusta varten. Hajautus parantaisi virhesietoisuutta ja mahdollistaisi myös tehoa vaativien työlastien jakamisen usealle eri tietokoneelle usean eri core-komponentin välille.

Resurssimanageri hallitsee pääsyä WSMX:n käyttämiin tietovarastoihin. Nykyversiossa rajapintoja on kuuteen eri tietosäiliöön: WSMO-objektit (web-palvelut, ontologiat, maalit ja välittäjät), muut tiedot (tapahtumat ja viestit) sekä WSDL-dokumentit, joita käytetään muodostettaessa WSMO-palvelukuvauksista niihin sopivia SOAP-viestejä.

Discovery-komponentti on vastuussa web-palveluiden etsimisestä. Web-palvelu valitaan ainoastaan, jos se on käyttäjän määrittelemän maalin kaltainen. Web-palveluiden etsimistä käsitellään enemmän kappaleessa 4.2.

Funktionaaliset ja ei-funktionaaliset valitsijat etsivät discovery-komponentin tuottamasta listasta kaikkien sopivimman palvelun. Funktionaalinen valitsin ei kutsu valittua palvelua, kun puolestaan ei-funktionaalinen kutsuu sitä valinnan jälkeen.

DataMediator-komponentti muuntaa tietoa suuntaan ja toiseen. Web-palveluiden etsimisessä, yhdistämisessä, valitsemisessä ja kutsumisessa voi esiintyä ongelmia erilaisten tiedostomuotojen eriäväisyyksien kanssa. DataMediator hakee tietosäiliöistä valmiiksi luodut kartoituksen eri muotojen välillä ja soveltaa niitä saapuviin ja lähteviin viesteihin.

Kommunikointimanageri on vastuussa eri tiedonsiirtoprotokollien käytöstä lähettäessä ja vastaanotettaessa tietoa WSMX:ssä. Viestejä otetaan vastaan Receiver-rajapinnan kautta ja web-palveluita kutsutaan Invoker-rajapinnan kautta. Invoker ottaa vastaan kutsuttua palvelua WSMO-kuvauksen ja sen tarvitsemat parametrit.

4.2 Web-palveluiden etsiminen

Web-palvelut voidaan WSMX:ssä etsiä sisäisistä tai ulkoisista säiliöistä. Yksi tapa etsiä palveluita on hakusanojen avulla. Hakusanat eivät käytä semanttisia tietoja, mutta niillä voidaan helposti ja nopeasti rajata tutkittavaa aineistoa [Kil05]. Hakusanat voidaan kohdistaa ontologian eri osiin, kuten esimerkiksi description-elementtiin,

jossa kuvataan ontologia vapaamuotoisesti. Avainsanahaku voidaan kohdistaa myös relaatioihin sekä maalien ja palveluiden kuvauksiin. Avainsanahaun ongelma on kuitenkin semantiikan puute. Merkkijonojen vertailulla ei mitenkään tietää esimerkiksi termien “VarausNumero” ja “ReservationNumber” tarkoittavan samaa asiaa. Palveluita voidaan etsiä myös niihin liittyvien käsitteiden perusteella. DataMediator-komponentti kartoittaa erilaisten ontologioiden eroavaisuudet ja yhtäläisyydet. Näin “VarausNumero” ja “ReservationNumber” voidaan linkittää toisiinsa, jolloin hakukone tietää niiden tarkoittavan samaa asiaa.

Nykyversiossa palveluiden etsintää hoitaa Discovery-komponentti, joka jakautuu useampiin pienempiin osiin. Discovery Manager on yksi sen alikomponenteista, joka pääosin vastuussa oikeanlaisen web-palvelun etsimisestä. Discovery Manager tekee ensin avainsanavertailun käyttäjän määrittelemän WSMO-maalin ja paikallisissa säiliöissä olevien palveluiden kuvausten välillä. Tuotettu lista suodatetaan edelleen palvelun aksioomiin ja WSMO-maaliin kohdistuvien semanttisten päättelyiden avulla. Suodatettu lista palautetaan kommunikointimanagerille, joka jatkaa edelleen oman ohjeistuksen mukaan palauttaen listan käyttäjälle tai lähettämällä sen selector-komponentille, joka valitsee listasta yhden palvelun suoritettavaksi.

4.3 Ulkoinen toiminta

WSMX:n toiminnallisuus voidaan kuvata sen tarjoamien pääsypisteiden (entry points) avulla [MZ05]. Pääsypisteet ovat rajapintoja, joiden kautta ulkopuolinen taho on yhteydessä WSMX:ään. Seuraavissa kappaleissa kuvataan kolme eri pääsypistettä (tai kutsua), joiden kautta käyttäjä voi pyytää palveluita WSMX:ltä.

Kunkin toiminnon kutsu aiheuttaa tapahtuman, joka laukaisee tietyn toimintasemantiikan. Pääsypisteet määrittelevät mitä WSMX tekee, kun puolestaan toimintasemantiikka kuvaa miten se tehdään [ZO05].

4.3.1 achieveGoal

WSMX etsii palvelun, valitsee sen ja suorittaa käyttäjän puolesta. Tätä varten käyttäjän on annettava parametrina WSML-dokumentti, jossa kuvataan tavoiteltava maali, ontologia ja optiot, joilla voi määrittellä esimerkiksi etsintäänkö palvelua paikallisista säiliöistä vai ulkoisilta palveluntarjoajilta. Toiminto edellyttää sen, että käyttäjä tietää ja on valmiiksi määritellyt kaikki parametrit, jota käytettävä palvelu tulee tarvitsemaan.

Toiminnon käynnistyessä discovery-komponentti luo ensiksi listan paikallisista ja ulkoisista web-palveluista. Luotu lista toimitetaan selection-komponentille, joka valitsee listasta parhaiten käyttäjän määrittelemää WSMO-maalia vastaavan palvelun. Palvelun käyttö saattaa edellyttää tiedonmuunnosta, johon käytetään data mediator-komponenttia. Kommunikointimanagerin invoker-komponentti kutsuu lopulta valittua palvelua. Kutsuttu palvelu voi tuottaa vastauksen, jonka Receiver-komponentti ottaa vastaan. Mikäli niin määritelty, lähetetään palvelunvastaus edelleen käyttäjän määrittelemään.

4.3.2 discoverWebServices ja invokeWebService

Toiminto tuottaa listan web-palveluista, jotka vastaavat käyttäjän määrittelemää WSMO-maalia. Lisäksi käyttäjän pitää antaa WSMX:lle parametreina käytettävän ontologian instanssi ja lisäoptiot (mm. kuinka monta palvelua palautetaan).

Kun käyttäjä on saanut WSMX:ltä listan sopivista web-palveluista, voi hän kutsua sitä WSMX kautta invokeWebService-toiminnolla. Parametreina käytetään ontologian instanssia, web-palvelun tunnusta ja käytettävän koreografian tunnusta. Koreografian avulla selvitetään palvelun käyttöön liittyvän viestinvälityksen eteneminen. Kuten achieveGoal-toiminnossa, WSMX palauttaa tarvittaessa palvelun tuottaman tuloksen käyttäjän määrittelemään osoitteeseen.

4.4 WSMX P2P-verkossa

WSMX ei ole suunniteltu toimimaan keskitettynä yksittäisenä järjestelmänä, vaan pikemminkin yhtenä P2P-verkon solmuna. P2P-verkot ovat aidosti hajautettuja verkkoja, jotka toimivat ilman keskitettyä palvelinta. Verkossa jokainen WSMX-ohjelmisto edustaa yhtä solmua. Solmuihin voi tarvittaessa kytkeä eri komponentteja, mikä tarkoittaa myös sitä, että yksittäisen solmun komponenttien ei tarvitse sijaita samalla palvelimella kuin komponenttia käyttävä WSMX.

P2P-verkon tuoma hyöty WSMX:ään liittyy web-palveluiden etsimiseen. WSMX:n saadessa maali-määrittelyn, se ensin vertaa sitä paikallisen palvelusäiliön sisältöön. Jos vastaavaa palvelua ei löydy, lähetetään määrittely edelleen seuraavalle solmulle ja sieltä vielä tarvittaessa eteenpäin. Palvelun etsimisen rekursio määritellään erikseen. P2P-verkossa WSMX-solmut toimivat kuten mitkä tahansa muutkin web-palvelut, eli niillä omat WSMO-määritelmänsä ja niitä kutsutaan samalla tavalla kuin muitakin web-palveluita.

5 Yhteenveto

WSMX on WSMO:n pohjalta toteutettu väliohjelmisto. Sen tarkoituksena on automatisoida semanttisten web palveluiden etsimiseen, valitsemiseen, koostamiseen ja suorittamiseen liittyviä operaatiota. WSMX:n nykyinen versio on hyvällä mallilla kulkemassa sen kehitystiimin ohjeistamaan suuntaan. Jo versionumero 0.5 kertoo yritysmaailmaan suunnatun kokonaisuuden olevan kuitenkin vielä jonkin matkan päässä, sillä esimerkiksi P2P-ominaisuus ei ole vielä lainkaan tuettu. Kehitysryhmän suunnitelmat kuulostavat kuitenkin varsin lupaavilta. Aidosti hajautettu väliohjelmisto P2P-verkossa voisi mahdollistaa uusien palveluiden nopean levittämisen ja hyödyntämisen.

Lähteet

- dBLK⁺05 de Bruijn, J., Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M. ja Fensel, D., *The Web Service Modeling Language WSMML*, 2005. URL <http://www.wsmo.org/TR/d16/d16.1/v0.21/>.
- FD05 Feier, C. ja Dominigue, J., *WSMO Primer*, 2005. URL <http://www.wsmo.org/TR/d3/d3.1/v0.1/20050401/>.
- Kil05 Kilgarriff, E., *WSMX Discovery Component*, 2005. URL <http://www.wsmo.org/TR/d13.12/v0.1/20051129>.
- MZ05 Micael Zaremba, Matthew Moran, T. H., *WSMX Architecture*, 2005. URL <http://www.wsmo.org/TR/d13/d13.4/v0.2/>.
- RLK06 Roman, D., Lausen, H. ja Keller, U., *Web Service Modeling Ontology (WSMO)*, 2006. URL <http://www.wsmo.org/TR/d2/v1.3/20061021/>.
- VMK⁺07 Vitvar, T., Mocan, A., Kerrigan, M., Zaremba, M., Zaremba, M., Moran, M., Cimpian, E., Haselwanter, T. ja Fensel, D., Semantically-enabled service oriented architecture : concepts, technology and application. *Service Oriented Computing and Applications*, 1,2(2007), sivut 129–154. URL <http://dx.doi.org/10.1007/s11761-007-0009-9>.
- W3C04 W3C, *Web Services Architecture*, 2004. URL <http://www.w3.org/TR/ws-arch/>.
- ZO05 Zaremba, M. ja Oren, E., *WSMX Execution Semantics*, 2005. URL <http://www.wsmo.org/TR/d13/d13.2/v0.2/>.