



Web-Pilarcos: väliohjelmistopalveluita sähköisille liiketoimintaverkostoille

Toni Ruokolainen, Janne Metso, Lea Kutvonen
Helsingin yliopisto

Tietojenkäsittelytieteen laitos

{Toni.Ruokolainen,Janne.Metso,Lea.Kutvonen}@cs.helsinki.fi

Tiivistelmä

Liiketoiminnan muuttuvat tarpeet luovat yritysten tietojärjestelmille vaatimuksia, joita ei tällä hetkellä yleisesti käytössä olevilla tekniikoilla ja teknologioilla pystytä tyydyttävästi täyttämään. Erityisesti yritysten välisen yhteistoiminnan tuki on puutteellista. Tässä artikkelissa esitellään web-Pilarcos -projekti, nykyaikaisten liiketoimintaverkostojen haasteiden ja vaatimusten pohjalta suunniteltu väliohjelmisto, joka piilottaa liiketoimintapalveluiden yhteentoimivuuteen ja liiketoimintaverkostojen luomiseen ja hallintaan liittyvät ongelmat varsinaisilta palvelutoteutuksilta.

1 Johdanto

Liiketoiminnan muuttuvat tarpeet luovat yritysten tietojärjestelmille vaatimuksia, joita ei tällä hetkellä yleisesti käytössä olevilla tekniikoilla ja teknologioilla pystytä tyydyttävästi täyttämään. Tietojärjestelmät ovat kankeita ja staattisia, eivätkä ne pysty tukemaan nopeasti muuttuvien liiketoimintaympäristöjen vaateita. Erityisesti yritysten välisen yhteistoiminnan tuki on puutteellista. Jotta nykyaikaisen liiketoiminnan tuomiin tekniisiin haasteisiin pystyttäisiin vastaamaan, tulee tietojärjestelmien toiminta perustaa liiketoiminnasta tuleviin vaatimuksiin.

Tekniikan kiivas kehitys eri toimialoilla on johtanut tilanteeseen, jossa yritysten ydinosaamisalueet ovat erikoistuneet voi-

makkaasti. Esimerkiksi auton- tai matkapuhelinvalmistajan ydinosaamisen alueet eivät kata läheskään koko sitä osaamista, jota tarvitaan kyseisten tuotteiden valmistamiseen. Autonvalmistajan onkin yleensä järkevämpää ja kustannustehokkaampaa tuottaa esimerkiksi autoon vaadittavat sulautetut tietojärjestelmät niihin erikoistuneella ulkopuolisella valmistajalla, kuin alkaa valmistaa niitä itse. Globalisaation seurauksena kilpailu yritysten kesken on voimistunut logististen rajoitusten hävittyä. Verkostoitumiskyky sekä yritysten muodostamien liiketoimintaverkostojen hallinta nousevat tärkeäksi osaksi yritysten liiketoimintastrategiaa ja kilpailukykyä.

Sähköiset liiketoimintaverkostot ovat yritysten tietojärjestelmien välisiä yhteis-

¹Termillä "liiketoimintaverkosto" viittaamme tästä lähtien nimenomaan yritysten välisen sähköisen lii-

toimintaverkostoja. Liiketoimintaverkoston¹ varsinaisia toimijoita ovat yritysten tuottamat sähköiset palvelut, jotka edustavat ulkoista käyttörajapintaa yrityksen tietojärjestelmiin. Eräs esimerkki liiketoimintaverkostosta on matkapaketin tilausverkosto, jossa on useita toimijoita: matkan tilaaja, matkatoimisto, sekä varsinaisen matkan toteuttavat toimijat kuten lentoyhtiö, hotelli ja autonvuokrausyhtiö.

Palvelupohjaisten liiketoimintaverkostojen muodostamiseen, niissä toimivien palveluiden yhteistyöhön ja verkoston ylläpitoon liittyy useita ongelmia ja haasteita. Ongelmat johtuvat siitä, että palvelupohjaiset liiketoimintaverkostot ja niissä toimivat palvelut ovat luonteeltaan heterogeenisia, toimijat ovat autonomisia ja ympäristö kokonaisuudessaan hyvin dynaaminen.

Helsingin yliopiston tietojenkäsittelytieteen laitoksella toimiva web-Pilarcos-projekti [5] (Production and integration of application systems with web technologies) tutkii ja kehittää sähköisten liiketoimintaverkostojen hallintamekanismeja sekä niissä käytettävien palveluiden ja verkostojen kuvausmenetelmiä. Projektin tavoitteena on luoda väliohjelmistoalusta palvelusuuntautuneiden liiketoimintaverkostojen käyttöön. Web-Pilarcos -projektiin käyttämä lähestymistapa on yhteneväinen sekä niin kutsutun *SOC-paradigman* [12] (Service Oriented Computing) että *ODP-viitemallin* [4] (Open Distributed Processing) kanssa.

Tämä artikkeli esittelee web-Pilarcos-projektin käsitteitä ja väliohjelmistopalveluita. Luvussa 2 analysoidaan tarkemmin nykyaikaisen, liiketoimintaympäristöön suunnatun väliohjelmiston tavoitteita ja tarpeita. Jotta nämä tarpeet voidaan täyttää, tulee liiketoimintaväliohjelmiston toteuttaa uudenlaisia palveluita, joita ei

ketoiminnan yhteistoimintaverkoston.

perinteisissä väliohjelmistoissa ole toistaiseksi käytetty. Nämä väliohjelmistopalvelut liittyvät metainformaation käsittelyyn ja hallintaan. Luvussa 3 perehdytään liiketoimintaverkostojen kuvaamisessa käytettäviin käsitteisiin, kuten liiketoimintaverkostomalleihin ja palvelutyyppeihin. Väliohjelmiston tärkeimmät palvelut kuvataan luvuissa 4, 5 ja 6, jotka käsittelevät vastaavasti metainformaation ja verkoston hallintaan liittyviä palveluita sekä liiketoimintaverkostojen suorituksenaikaista hallintaa.

2 Liiketoimintaverkostojen haasteet

Yritykset käyttävät tietojärjestelmiä tehostaakseen toimintaansa ja edistääkseen omien liiketoimintaprosessiensa kontrolloitavuutta ja läpinäkyvyyttä. Nykyaikaisessa yrityksessä onkin useita erilaisia järjestelmiä, joilla on erilaiset tavoitteet ja roolit. Esimerkiksi tuotekehitykseen, asiakastietohallintaan ja varastohallintaan kehitettyjen ohjelmistojen kohderyhmät ovat hyvinkin erilaiset. Liiketoiminnan tehostamisen nimissä nämäkin hyvin erilaisiin tarpeisiin suunnatut tietojärjestelmät tulisi saada yhdistettyä toimivaksi kokonaisuudeksi.

Liiketoimintaan liittyvien tietojärjestelmien integrointiin suuremmiksi ja kokonaisvaltaisemmiksi järjestelmiksi käytetään erilaisia teknologioita. Yrityksen sisäisten tietojärjestelmien integrointiin voidaan käyttää esimerkiksi *väliohjelmistoja* tai *EAI-* (Enterprise Application Integration) ja *ERP-* (Enterprise Resource Planning) alustoja. Väliohjelmistoihin kuten Java J2EE [13] tai OMG:n CORBA [9], perustuvassa integroinnissa kaikki yrityksen liiketoimintajärjestelmät on

toteutettu samalle ajonaikaiselle alustalle. Yhteinen toiminta-alusta yleensä takaakin järjestelmien yhteentoimivuuden teknisellä tasolla.

Erityiset EAI- ja ERP-alustat on kehitetty suurehkojen yritysten tarpeisiin, joissa esimerkiksi yritysostojen ja -integraatioiden vuoksi ei yhtenäisiä toteutuslustoja tai liiketoimintaprosesseja ole välttämättä käytettävissä. EAI-alustat sisältävät teknologioita ja tekniikoita, joiden avulla yrityksen olemassa olevia järjestelmiä (legacy system) voidaan integroida toisiinsa yhtenäiseksi kokonaisuudeksi. EAI-järjestelmä voi esimerkiksi tukea uusien liiketoimintaprosessien muodostamista vanhojen järjestelmien toiminnallisuutta hyödyntäen. ERP-lähestymistavassa taas "pakotetaan" vanhat järjestelmät uuteen muotoon käyttämällä integrointialustana ERP-järjestelmää, joka tarjoaa joukon ennalta määrättyjä liiketoimintaprosesseja yrityksen käytettäväksi [6].

Nykyaikaisessa verkottuneessa liiketoimintaympäristössä yritysten sisäisten tietojärjestelmien integrointi ei kuitenkaan enää riitä kilpailukykyisen liiketoiminnan takeeksi. Sisäisen integraation lisäksi tarvitaan myös kykyä liittyä muiden yritysten tietojärjestelmiin. Tällainen yritysten välinen yhteistyö tuo mukanaan aivan uudenlaisia haasteita, joihin ei pystytä vastaamaan perinteisillä integrointimenetelmillä.

Yksi yleisimmistä esteistä yritysten väliselle yhteistyölle on monimuotoisuus, joka ilmenee sekä teknisellä että semanttisella tasolla. Teknisellä tasolla monimuotoisuutta eli heterogeenisyyttä aiheuttavat yritysten tietojärjestelmien erilaiset toteutusmallit ja -tekniikat. Teknologisen heterogeenisyyden poistaminen on välttämätön ehto tietojärjestelmien yhteistoiminnalle ja se saavutetaan yleensä hyödyn-

tämällä yhtenäistämistekniikoita ja -teknologioita, kuten ohjelmistoadaptereita ja väliohjelmistotalustoja. Ohjelmistoadapterit ovat sovellusten välille toteutettuja välittäjäkomponentteja, jotka suorittavat sovelluskohtaisia tiedonmuunnoksia esitysmuodosta toiseen. Väliohjelmistotalustat, kuten CORBA [9] tai J2EE [13], tarjoavat sovellusten käyttöön esimerkiksi yhtenäiset kommunikointirajapinnat, tiedonmuunnospalvelut, laskentamallin (esimerkiksi objekti- ja komponenttimallit) sekä yleisesti tarvittuja tukipalveluita.

Semanttista heterogeenisyyttä ei voida poistaa suoraviivaisilla muunnos- tai yhtenäistämisooperaatiolla, vaan kyse on perustavanlaatuisesta eriäväisyydestä tavoissa mallintaa maailmaa kussakin yrityksessä. Semanttista monimuotoisuutta voi ilmetä usealla eri tasolla ja alueella liiketoimintaverkostossa. Esimerkiksi informaation tasolla lukuarvot voidaan tulkita eri tavalla: ruuveja tilattaessa kappalemäärä tarkoittaa yksittäisten ruuvien määrää toisessa yrityksessä, kun taas toisessa yrityksessä ruuveja käsitellään paketeittain. Ylimmällä tasolla heterogeenisyyttä ja siitä johtuvia tulkintavirheitä voi ilmetä esimerkiksi erilaisista liiketoimintastrategioista ja -säännöistä johtuen.

Osa yritysten välisen yhteistoiminnan ongelmista johtuu siitä, että yksittäiset yritykset ovat olennaisesti autonomisia toimijoita. Yrityksen autonomialla tarkoitetaan sitä, että yritys kontrolloi omia informaatiojärjestelmiään ja tarjoamiaan liiketoimintapalveluita. Yksittäinen yritys muodostaa siis autonomisen hallintoalueen liiketoimintaverkostossa. Toisaalta, yritykset toteuttavat autonomisia intentioitaan. Tämä tarkoittaa sitä, että yrityksillä on omat strategiansa ja toimintatapansa, joihin muut eivät pysty vaikuttamaan. Nämä autonomiset intentiot vaikuttavat osaltaan siihen, kuinka yrityksen

tarjoamia liiketoimintapalveluita voidaan käyttää. Yritykset myös tekevät itsenäisiä ratkaisuja siitä, minkälaisiin liiketoimintaverkostoihin ne liittyvät ja millä ehdoilla.

Liiketoimintaverkostot ovat luonteeltaan dynaamisia: uusien liiketoimintamallien myötä tarvitaan uudenlaisia liiketoimintaverkostoja ja jo käytössä olevia liiketoimintaverkostoja täytyy päivittää aika ajoin vastaamaan todellisia tarpeita. Myöskään palveluntarjoajien tuottamien palveluiden joukko ei ole muuttumaton, vaan uusia palveluita tulee esille samalla kun vanhentuneita palveluita poistetaan tai muutetaan. Yritysten autonomisuus aiheuttaa myös osaltaan dynaamisuutta ympäristöön, sillä tarjottujen palveluiden ominaisuudet voivat vaihtua varoittamatta. Nykyaikaisessa verkottuneessa liiketoiminnassa yritysten tietojärjestelmiltä vaaditaan myös kykyä sopeutua suhteellisen nopeasti muuttuviin tarpeisiin. Lisäksi yritysten pitää pystyä toiminaan samaan aikaan useassa eri liiketoimintaverkostossa.

Sähköisen liiketoiminnan tarpeisiin tarkoitettun ympäristön tulee tukea liiketoiminnassa käytettäviä toimintamalleja ja käsitteistöä. *Liiketoimintaprosessi* on luontainen toimintamuoto liiketoimintaympäristöissä. Liiketoimintaprosessi kuvaa tietyn liiketoimintaskenaarion toiminnan määrittelemällä joukon aktiviteetteja sekä järjestyksen näiden aktiviteettien kesken. Liiketoimintaprosessilla voidaan kuvata esimerkiksi informaation kulkua toimitusketjussa. Keskeisiä käsitteitä ovat myös liiketoimintastrategiat ja -säännöt sekä niin kutsutut *politiikat*, jotka säätelevät liiketoimintaverkoston toimintaa lisämäärein ja -määräyksin.

Kaikilla edellä mainituilla liiketoimintaverkostojen ominaisuuksilla on se vaikutus, että liiketoimintaa tukevien tieto-

järjestelmien *yhteentoimivuus* muodostuu ongelmalliseksi. Yhteentoimivuudella tarkoitetaan yksinkertaistaen sitä, että kaksi erillistä järjestelmää pystyvät kommunikoimaan keskenään siten, että informaatio tulee ymmärretyksi molempien toimijoiden puolella "samalla tavalla" ja että näiden toimijoiden yhteistyö johtaa yhteisen päämäärän saavuttamiseen.

Yhteentoimivuus ollaan perinteisesti saavutettu järjestelmien *tiukalla integroinnilla*. Tässä toteutusmallissa välttämätön informaatio yhteentoimivuuden takaamiseksi on kudottu ohjelmistototeutuksien osaksi ohjelmoijien toimesta. Tämä toteutusmalli ei siedä juuri lainkaan liiketoimintaverkostojen ja -palveluiden evoluutiota tai dynaamisuutta, sillä toiminnallisuuden muutoksiin ei pystytä sopeutumaan. Toisaalta, jos voidaan olla aivan varmoja siitä, että kohdeympäristö pysyy muuttumattomana toteutushetkestä aina järjestelmän alasajoon asti, niin tiukka integrointi voi olla hyvinkin kustannustehokas yhteentoimivuuden tae. Suurin osa yritysten sisäisten tietojärjestelmien ja yritysten välisistä yhteentoimintaratkaisuksista on toteutettu tällä hetkellä tiukkaa integrointimallia käyttäen.

Hieman joustavampi yhteentoimivuuden toteutusmalli saadaan käyttämällä *yhtenäistämistä* yhteiseen malliin (unification). Tässä toteutusmallissa liiketoimintapalvelut generoidaan yhteisen mallin, kuten standardin tai UML-kaavion, pohjalta. Yhteentoimivuudelle välttämätön informaatio on siis osa yhteistä mallia, josta se koodin generoinnin yhteydessä siirtyy osaksi ohjelmistoa. Liiketoimintaympäristön muutoksiin sopeudutaan tässä mallissa siten, että uudesta liiketoimintaympäristöstä tehdään uusi malli, jonka perusteella generoidaan uudet liiketoimintapalvelut. Vaikka tämä malli tukeekin paremmin liiketoimintaympäristön

muutoksia, ei se kuitenkaan siedä liiketoimintaverkostojen dynaamisia muutoksia muuten kuin niissä tapauksissa, missä kyseiset muutokset ja niihin reagoinnit ovat osa yhteistä mallia. Esimerkiksi niin kutsuttu malliperustaiseen arkkitehtuuriin (Model Driven Architecture, MDA) [3] perustuva sovelluskehitys sekä ebXML- ja Rosettanet-yritystoimintastandardit [2, 11] voidaan lukea tätä yhteentoimivuusmallia toteuttaviksi tekniikoiksi.

Koska sekä liiketoimintaverkostot että niissä käytettävät liiketoimintapalvelut ovat ”liikkuvia maaleja”, eivät edellä kuvatut perinteiset mallit yhteentoimivuuden takaamiseksi ole toimivia. Liiketoimintapalveluiden *löyhään liittoumaan* (federation) perustuvassa liiketoimintaverkostossa palveluiden yhteentoimivuus taataan neuvotteluiden, yhteentoimivuustarkastuksien ja ajonaikaisen valvonnan avulla. Löyhän liittouman muodostamisessa hyödynnetään liiketoimintaverkostoja ja -palveluita kuvaavaa metainformaatiota, joka on julkistettu liiketoimintaväliohjelmiston metainformaatiovarastoihin. Näiden kuvausten perusteella päätetään, mitkä palvelutarjoukset voidaan ottaa mukaan verkostoon ja millä parametreilla. Metainformaatiota käytetään yhteentoimivuustarkastuksiin verkoston muodostamisen yhteydessä ja yhteisön hallintaan liiketoimintaverkoston toiminnan aikana.

3 Käsitteistö

Web-Pilarcos -projektissa [5] kehitetään niin kutsuttua B2B-väliohjelmistoa (business-to-business) nykyaikaisen verkottuneen liiketalouden tarpeisiin. Näkemysksemme mukaan liiketoimintaverkostojen sekä siinä toimivien palveluiden haasteisiin ja vaatimuksiin voidaan vastata parhaiten käyttämällä löyhän liit-

touman mallia. Tämä yhteistoimintamalli tarvitsee toimiakseen hyvin määritellyn käsitteistön, johon perustuen liiketoimintaverkostot ja -palvelut voidaan mallintaa. Näiden käsitteiden tulee erityisesti palvelusta sekä liiketoiminnan että ohjelmistotuotannon tarpeita.

Liiketoimintapalvelu (tästä lähtien yksinkertaisesti ”palvelu”) on itsensä kuvaava ja itsenäinen ohjelmistokomponentti, joita koostamalla voidaan toteuttaa monimutkaisempia ohjelmistoja [12]. Palvelu on käytännössä ohjelmistokomponentti, joka tuottaa jonkin liiketoimintaan liittyvän toiminnallisuuden hyvin määritellyn rajapinnan kautta. Palvelu on yksirajapintainen, jolla tarkoitetaan sitä että jokainen palvelu tuottaa tietyn toiminnallisuuden vain yhdelle asiakkaalle kerrallaan. Tämän rajoituksen avulla halutaan välttää yksittäisen palvelun turhaa erikoistamista tiettyyn liiketoimintamalliin, siten edistään palvelun modulaarisuutta ja uudelleenkäyttöä eri verkostoissa. Palveluiden yhdistäminen liiketoimintarooleiksi kuvataan liiketoimintaverkoston arkkitehtuurissa. Sähköisissä liiketoimintaverkostoissa palvelut voidaan toteuttaa esimerkiksi Web-palvelut -teknologiaa käyttäen [18]. Olennaisin osa palvelun käsitettä ovat palvelukuvaukset, jotka määrittelevät palveluiden toiminnallisuudet sekä niiden parametrit. Palvelukuvauksia on kahta erilaista luokkaa, palvelutyyppejä ja palvelutarjouksia.

Palvelutyyppi on abstrakti palvelukuvaus, joka kuvaa miten tietyn tyyppisen palvelun tulee toimia sekä minkälaisia ominaisuuksia sillä tulee olla. Palvelutyyppi on olennaisesti samankaltainen ohjelmointikielistä tutun tyyppin käsitteen kanssa: se takaa tyyppiturvallisuuden ja sitä käytetään luokittelemaan universuminsa alkiota, tässä tapauksessa palveluita, eri luokkiin niiden ominaisuuksien mu-

kaan. Tyypiturvallisuudella tarkoitetaan yksinkertaista tyypityksen luomaa ”turvallisuustaetta”: jos palvelu on oikein tyypitetty, niin silloin palvelun käytöstä ei tulisi seurata mitään ongelmia. Palveluiden luokittelu taas suoritetaan tyyppijärjestelmässä määriteltyjen tyyppiekvivalenssi- ja alityypitysmääritysten mukaisesti.

Palvelutyypin määrittelee palvelun rajapinnan rakenteen (syntaksin) ja toimintasemantiikan. Tärkeimpänä osana toimintasemantiikkaa palvelutyypissä määritellään niin kutsuttu *rajapintaprotokolla*. Rajapintaprotokolla on prosessikuvaus, joka määrittelee palvelun ulospäin näkyvän käyttäytymisen viestien kommunikoinnin tasolla. Se siis määrittelee minikälaisia viestejä palvelu odottaa saavansa ja mikä on niiden keskinäinen järjestys. Rajapintaprotokollan kuvaus annetaan XML-pohjaisella [17] prosessinkuvauskielellä, jonka formaali semantiikka perustuu π -kalkyylinä tunnettuun prosessialgebraan. Kyseinen prosessialgebrojen perhe sisältää käsitteitä ja todistustekniikoita, joita voidaan soveltaa sekä tyypiturvallisuuden (yhteentoimivuuden) tarkaamiseen että palvelutarjousten luokittelamiseen.

Toinen osa palvelutyypissä määriteltävää toimintasemantiikkaa liittyy kommunikoitavien viestien sisällön tulkintaan. Koska samanlainen syntaktinen muoto kahden viestin välillä ei vielä takaa, että niiden tarkoitettu tulkinta on sama, tarvitaan mekanismeja, joiden avulla viesteihin voidaan liittää yksikäsitteinen tulkinta. Eräänlainen tulkintamekanismi saadaan aikaan hyödyntämällä julkisia ontologioita. Ontologia on jonkin toimialueen käsitteistön kuvaus, joka määrittelee käsitteille rajallisen termistön sekä termien väliset suhteet. Ontologisten käsitteiden avulla voidaan esimerkiksi erottaa toisistaan kuvaako tietty kokonaislu-

ku dollareita vai euroja. Ontologioita käytetään tällä hetkellä yleisesti niin kutsutuissa Semantic Web -ympäristöissä, joissa niiden kuvaamiseen on kehitetty XML-pohjaisia ontologioiden kuvauskieliä, kuten OWL [15] (Web Ontology Language).

Palvelutyypin yksinkertaistettu kuvaus esitellään kuvassa 1. Siitä voidaan tunnistaa palvelutyypin tärkeimmät osiot, joita ovat palveluattribuutit (elementti *serviceAttributes*), dokumenttityyppien kuvaukset (elementti *documentTypes*) ja käyttäytymiskuvaus (elementti *iproto*).

Palveluattribuutit on määritelty nelikona (nimi, tyyppi, konsepti, rajoite), joista kaksi ensimmäistä on pakollisia, loput vapaaehtoisia. Konsepti sitoo attribuuttiin ontologisen käsitteen, kun taas rajoite määrää kyseisen tyyppiselle muuttujalle sille sallitun arvojoukon.

Kommunikoinnissa käytettävien viestien tyypit määritellään elementissä *documentTypes*. Dokumenttityyppien määrittelyyn käytetään XML-Schema-standardin mukaisia rakenteita [16]. Lisäksi valmiiksi määriteltyjä dokumenttityyppejä voidaan sisällyttää elementtiin *import-lauseella*; tällöin tyyppimäärittelyt haetaan nimiavaruuden tunnisteiden avulla kyseistä nimiavaruutta hallinnoivasta tyyppivarastosta. Tyyppivarasto on eräs metainformaation hallintapalveluista, jotka esitellään tarkemmin luvussa 4.

Palvelutyypin käyttäytyminen määritellään rajapintaprotokollana, joka kuvaa palvelun ja sitä käyttävän asiakkaan välisen yksittäisen istunnon käyttäytymisen. Rajapintaprotokollan kuvaava elementti *iproto* sisältää varsinaisen käyttäytymiskuvauksen sekä käyttäytymiskuvauksessa käytettävien kommunikointikanavien määrittelyjä. Käyttäytymiskuvaus annetaan elementissä *session*, jossa peräkkäisistä kommunikointiprimitiiveistä (viestin

```

<serviceType
  name = xs:string>
(serviceAttributes?, documentTypes, iproto ) <iproto
  name = xs:NCName>
(channel+, session)
</serviceType>
</iproto>
<serviceAttributes>
(property+)
</serviceAttributes>
<channel
  name = xs:NCName>
[channel properties]
</channel>
<property
  name = xs:string
  type = xs:QName
  concept = xs:QName?>
constraint?
</property>
<session
  name = xs:NCName>
(parallel | sequence |
if | repeat | choice )?
</session>
<documentTypes>
[XML-Schemas for documents]
</documentTypes>

```

Kuva 1: Palvelutyyppin yksinkertaistettu kuvaus.

lähetyt ja vastaanotto) koostettuja prosesseja voidaan edelleen yhdistää rinnakkaisiksi (parallel), valinnaisiksi (choice), toistetuiksi (repeat) ja ehdollisiksi (if) prosesseiksi. Kanavamäärittelyt annetaan joukkona channel -elementtejä. Kanavamäärittelyyn voidaan liittää esimerkiksi määreitä tietoturvan tasosta tai siitä, voidaananko kanavaa käyttää käyttäytymiskuvauksessa uudelleen.

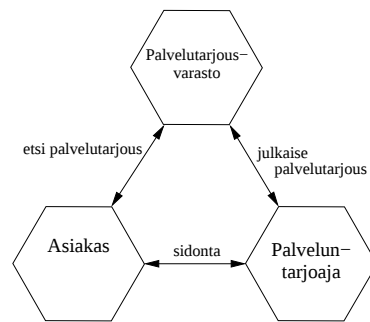
Palvelutarjous on palveluntarjoajan tuottama kuvaus, joka määrää arvot palvelutyypissä vaadituille parametreille. Palvelutarjoukset edustavat varsinaisia palveluinstansseja ja ne sisältävät palvelutyyppin kuvaamien ominaisuuksien lisäksi tiedon siitä, kuinka varsinaiseen palveluun pääsee teknisesti käsiksi. Palvelutarjouksia julkaistaan palvelutarjousvarastoihin, joista potentiaaliset asiakkaat (muut yritykset ja toimijat) voivat niitä etsiä palvelutyyppin ja parametriarvojen perusteella. Kun sopiva palvelutarjous on löytynyt, voidaan asiakkaan ja palvelun tarjoajan välille muodostaa sidonta. Palveluprustaisen ympäristön yleinen toimintamalli on esitelty kuvassa 2.

Kuvausta, joka esittää kokonaisen liiketoimintayhteisön rakenteen ja toiminnalliset vaatimukset kutsutaan *liiketoimintaverkostomalliksi*.

Rakenne, tai arkitekhtuuri, määrittellään *liiketoimintaroolien* (roolien) ja niiden välisten sidontojen kautta. Rooli määrää liiketoimintayhteisön toimijan ominaisuudet sitomalla yhden tai useamman palvelutyyppin kokonaisuudeksi. Jokainen palvelutyyppi edustaa roolissa yhtä palveluporttia, johon jonkin muun roolin portti voidaan sitoa. Rooleissa esiintyy kahta erilaista palvelukompositiotyyppiä, joita kutsutaan toiminnalliseksi ja hallinnolliseksi palvelukompositioksi.

Toiminnallisessa palvelukompositiossa roolin yksittäisissä palvelutyypeissä esiintyvien rajapintaprotokollien välille on määritelty yksi tai useampi koordinoitietiehto. Koordinointietiehto voi olla esimerkiksi muotoa “portista A tulee olla saatuna viesti ’Maksu’, ennen kuin portista B voidaan lähettää eteenpäin viesti ’Kuittaus’”. Palvelutyyppien mukaiset rajapintaprotokollat ja roolin koordinoitietiedot yhdessä muodostavat roolin toteuttaman liiketoimintaprosessin.

Hallinnollisessa palvelukompositiossa puolestaan palveluita halutaan yhdistää sillä perusteella, että niillä on jonkinlainen reaali maailmasta tuleva vaade, joka sitone yhteen. Tällainen vaade voisi esimer-



Kuva 2: Palveluperustaisen järjestelmän perusarkkitehtuuri.

kiksi määrätä, että roolin toteuttavan toimijan tulee tarjota sekä “TilillePano”- että “SaldoKysely”-palvelutyyppit.

Roolien väliset sidonnat ovat palveluporttien välisiä yhteyksiä, missä jokainen roolin palveluportti edustaa erillistä yhteyttä jonkin toisen roolin kanssa. Lisäksi sidonnoissa voidaan määritellä yhteyksien yleisiä ominaisuuksia, kuten viestinnän tietoturvaan liittyviä attribuutteja. Roolien ja niiden välisten sidontojen lisäksi liiketoimintaverkkomallissa voidaan määritellä lisäominaisuuksia yhteisölle, esimerkiksi rooli- ja yhteisökohtaisia lisärajoitteita palvelun laatutason suhteen, luottamukseen liittyviä määreitä tai muita ei-toiminnallisia lisäpiirteitä.

Toimijoiden välinen yhteistoiminta sähköisessä liiketoimintaverkostossa on web-Pilarcos -ympäristössä sopimuspohjaista [5]. Liiketoimintayhteisön muodostamisen yhteydessä luodaan liiketoimintaverkostomalleihin perustuen niin kutsutuja *sopimuksia*. Sopimukset ovat yhteisön osallistujien välisiä ja yhdessä muodostamia sitoumuksia, joissa eritellään millä palveluparametrien arvoilla toimijat lupaavat osallistua yhteisön toimintaan. Jokainen yhteisön toimija valvoo yhteisön toiminnan aikana, että sopimuksessa luovuttu yhteistyön muoto tulee täytettyä.

4 Metainformaation hallintapalvelut

Web-Pilarcos -väliohjelmiston toiminta perustuu palveluita ja yhteisöjä kuvaavan metainformaation hyödyntämiseen. Suurin osa väliohjelmistopalveluista liittyykin suoraan metainformaation käsittelemiseen ja hallitsemiseen. Web-Pilarcos -arkkitehtuurista voidaan tunnistaa kolme erilaista infrastruktuuripalveluiden luokkaa: 1) metainformaatiovarastot, 2) verkostonhallintapalvelut ja 3) valvontakoneisto. Metainformaatiovarastot muodostavat julkisen hajautetun infrastruktuurin, jonka avulla liiketoimintaverkostojen ja -palveluiden kuvauksia voidaan julkaista ja hakea. Verkostonhallintapalvelut liittyvät liiketoimintaverkoston ajonaikaiseen hallintaan. Valvontakoneistoa tarvitaan osana verkostonhallinnan toteuttamiseksi. Tässä luvussa kuvaillaan metainformaatiopalveluiden toimintaa. Verkostonhallintamekanismeihin tutustutaan luvussa 5 ja yhteisön ajonaikaiseen valvontaan perehdytään luvussa 6.

Metainformaatiovarastoihin kuuluu viisi julkista väliohjelmistopalvelua. Niin kutsuttu *tyyppivarasto* hallitsee tyyppitietoa, kuten palvelutyyppikuvauksia, viesteissä käytettävien tietotyyppien kuvauk-

sia sekä näiden tyyppien välisiä suhteita. Tyyppivarasto ylläpitää kuvauksien muodostamia tyyppihierarkioita, esimerkiksi alityypitysuhteita palvelutyypin välillä, ja se myös antaa tyyppikuvauksille yksikäsitteiset nimet. Erityisesti tyyppivarasto huolehtii palvelukuvausten tyyppiturvallisuudesta toteuttamalla tyyppitarkastuksia uusien kuvauksien julkaisujen yhteydessä.

Globaalisti yksikäsitteisten nimien hallintaa ja nimien käyttöön perustuvaa resurssien etsintää varten tarvitaan *nimipalvelua* (name registry), jonka kautta tiettyä nimiavaruutta tai nimeä ylläpitävä taho, esimerkiksi juuri tyyppivarasto, voidaan löytää. Nimipalvelua voidaan verrata yleisesti tunnettuun Internetin nimipalveluun [7], DNS:ssään. Toisin kuin DNS:ssä, jossa sidotaan nimiä IP-osoitteille, web-Pilarcos -väliohjelmiston nimipalvelu nimeää metainformaatioresursseja ja sitoo nimet niitä hallinnoiviin elimiin. Nimentään käytetään URI-standardin [1] mukaisia nimiä ja kyseisen standardin mukaista nimiavaruuksien käsitettä käytetään antamaan nimien joukolle rakennetta.

Liiketoimintaverkoston kuvauksia säilytetään ja hallitaan *BNM-varastossa*. Tämä metainformaatiovarasto on toiminnaltaan melko yksinkertainen, toteuttaen vain tarvittavat varastointi-, haku-, ja ylläpitotoiminnot.

Palvelutarjousvarastoa käytetään varsinaisten palvelutarjousten julkaisemiseen ja etsimiseen. Palvelutarjousvarasto toteuttaa ”keltaiset sivut” -tyyppisen julkisen palvelun, jonka avulla palveluntuottajia voidaan etsiä palveluiden ominaisuuksien, erityisesti tyyppin, perusteella. Palvelutarjousvarasto käyttää hyväkseen tyyppivaraston palveluita toteuttaakseen tyyppitarkastuksia uusien palvelutarjousten julkaisujen yhteydessä.

Edellä mainittujen julkisten metainformaatiovarastojen lisäksi jokaisella yhteisön jäsenellä voi olla käytössään niin kutsuttu *politiikkavarasto*, jossa ylläpidetään kyseisen organisaation toimintapolitiikkojen kuvauksia. Toimintapolitiikat voivat olla joko julkista tai yksityistä tietoutta riippuen kyseisen toimijan intresseistä. Toimintapolitiikat määrittelevät yrityksen sisäisiä säännöstöjä, joita sovelletaan liiketoimintaverkostoissa toimitaessa.

Metainformaatiovarastojen roolia ja käyttöä liiketoimintaverkoston muodostamisen yhteydessä on mallinnettu esimerkinomaisesti kuvassa 3. Palvelun tarjoaminen on esitelty kohdissa 1)–4). Erityisesti huomioitavaa prosessissa on kohta 3, jossa tyyppivaraston tietoutta käyttäen tarkastetaan palvelutarjouksen ja sitä vastaavan palvelutyypin yhteensopivuus.

Kohdat A)–E) kuvaavat liiketoimintaverkoston muodostamisessa tarvittavia osatehtäviä. Liiketoimintaverkoston muodostaminen alkaa siten, että toiminnalle sopiva liiketoimintaverkoston kuvaus haetaan BNM-varastosta alkuunpanijan toimesta (kohta A). Tämän jälkeen toiminta siirretään julkiselle *populaattoripalvelulle*, jonka tehtävänä on täyttää eli populoida haluttu verkostomalli yhteensopivilla palvelutarjouksilla.

Populaattori hakee tyyppivarastosta liiketoimintaverkostomallin rooleihin yhteensopivia palvelutyyppejä (kohta B). Yhteensopivien palvelutyypin haussa voidaan roolissa määritellyn palvelutyypin sijasta etsiä tyyppivarastosta saman kaltaisia palvelutyyppejä. Tällöin palvelutyypin välillä on oltava voimassa niin kutsuttu *sovitusrelaatio* (matching), joka tarkoittaa että kyseiset palvelutyypit ovat ”tarpeellisella tasolla” saman kaltaiset. Sovitusrelaatiot annetaan palveluiden julkaisijoiden puolesta ja nii-

hin voi liittyä esimerkiksi yksinkertaisia tiedonmuunnostarpeita. Tiedonmuunnos voidaan tarjota esimerkiksi erillisenä palveluna tai jos kyse on yksinkertaisista XML-pohjaisten tietorakenteiden välisistä muutoksista, niin muunnos voidaan antaa XSLT-skriptinä [14]. Muunnosten oikeellisuutta ei voida automaattisesti tarkistaa, joten niiden julkaisua ja käyttöä tulee voida rajoittaa hallinnollisilla toimilla.

Palautettujen palvelutyyppien perusteella haetaan palvelutarjousvarastosta tyypejä vastaavia palvelutarjouksia (kohta C). Saatuaan tarvittavat palvelutarjoukset, populaattori suorittaa rajoitekarsinnan, jossa toisiensa kanssa parametrien suhteen ristiriitaiset palvelutarjoukset pudotetaan verkostosta pois (kohta D).

Populoinnista saadaan joukko ehdotuksia liiketoimintayhteisöiksi. Näihin yhteisöehdotuksiin sovelletaan vielä toimijoiden välisiä neuvotteluita lopullisista yhteisön parametreista. Neuvotteluprosessi on kuvattu tarkemmin luvussa 5. Onnistuneen neuvottelun tuloksena saadaan yhteisösopimus, jossa eritellään yhteistoiminnan lopullinen muoto. Yhteisösopimuksessa pitäytymistä tarkkaillaan verkoston toiminnan aikana jokaisen osallistujan osalta.

5 Verkostonhallintamekanismit

Väliohjelmiston verkostonhallintapalvelut jakautuvat kahteen osaan. Nämä ovat *verkostonhallinta-agentti* ja *sopimusolio*. Verkostonhallinta-agentin tehtävänä on edustaa organisaatiota liiketoimintaverkostoissa. Jokaisella organisaatiolla on vähintään yksi agentti. Agentti vastaa liiketoimintaverkoston hallintaan liittyvien protokollien suorituksesta. Sopimusolion tehtävänä on edustaa verkostoa organi-

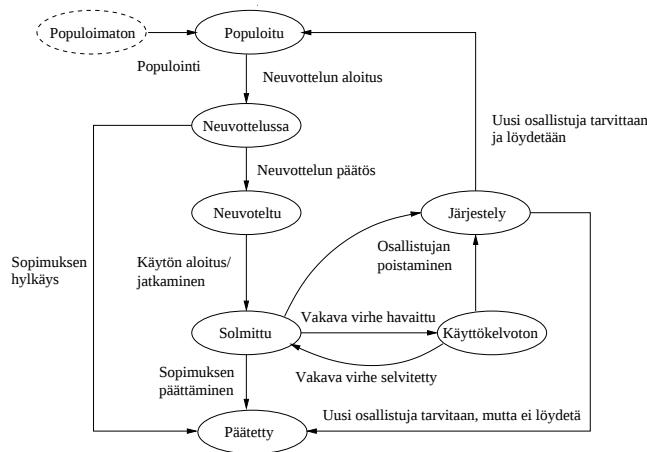
saatiossa. Se toimii sopimukseen liittyvien tietojen varastona mutta myös itsenäisenä päätöksentekijänä sopimukseen liittyen. Oliolla on oma tila, joka vastaa liiketoimintaverkoston sopimuksen elinkaaren vaihetta. Liiketoimintaverkoston tilaa ylläpidetään jokaisessa organisaatiossa erikseen. Tilan ylläpitämiseen osallistuu agentti välittämällä tilatietoa organisaatioiden välillä; sopimusolio ylläpitämällä sopimuksen elinkaaren vaihetta; sekä seuraavassa kappaleessa esiteltävä valvontakoneisto.

Verkostonhallinta-agentilla on hallintaprotokollat jokaiselle liiketoimintaverkoston elinkaaren vaiheelle ja sopimusrikkomusten hallintaan. Sopimus pohjaisen verkoston elinkaareissa on neljä merkittävää vaihetta. Nämä ovat verkoston pystytys, neuvottelu, verkostoon liittyvien toimintaprosessien suorittaminen ja verkoston alasajo toiminnan jälkeen [8]. Web-Pilarcos -prototyypissä jokaista vaihetta kohden on yksi tai useampia hallintaprotokollia, joita verkostonhallinta-agentti suorittaa tarpeen mukaan.

Web-Pilarcos -prototyypissä verkoston elinkaari on jaettu useampiin vaiheisiin kuin edellä on mainittu. Tämä johtuu suoritusaikaisen ympäristön tarpeista ja siihen liittyvistä luonnollisista vaiheista.

Kuvassa 4 esitellään web-Pilarcos -prototyypissä käytetyt sähköisen liiketoimintaverkoston elinkaaren vaiheet. Verkoston elinkaari alkaa populoinnista jolloin populaattorilta pyydetään yksi tai useita joukkoja yhteensopivia palvelutarjouksia verkostomallin perusteella. Populaattorin vastauksena antamista joukoista valitaan yksi neuvottelua varten.

Populoinnin jälkeen sopimusolio luodaan ja sen tilaksi alustetaan *Populoitu*. Neuvotteluiden alkaessa verkosto siirtyy *Neuvottelussa*-tilaan. Tämä tarkoittaa si-



Kuva 4: Verkoston elinkaaaren vaiheet.

taan, verkosto siirtyy *Järjestely*-tilaan jonka seurauksena haetaan uutta toimijaa verkostoon. Jos uusi toimija löytyy, siirytään takaisin neuvotteluun jotta sopimuksen yksityiskohdissa voidaan ottaa huomioon uusi verkoston osallistuja. Jos taas uutta osallistujaa ei löydetä, verkoston toiminta päättyy ja verkosto siirtyy *Päätetty*-tilaan.

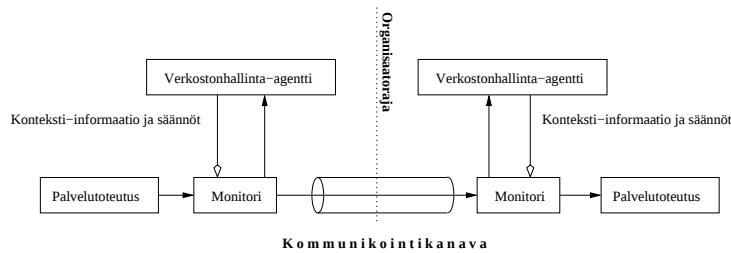
Jos sopimuksen neuvottelu epäonnistuu, eli jokin osapuoli lopettaa neuvottelut kesken, sopimuksen muodostus keskeytyy. Näissä tilanteissa voidaan valita uusi yhteisöehdotus populaattorilta saadusta joukosta. Jos kaikki neuvottelut epäonnistuvat, joudutaan suorittamaan uusi populointikutsu.

6 Valvontamekanismit

Valvontakoneiston tehtävänä on seurata palveluiden toimintaa liiketoimintaverkostossa sovitun palveluiden käyttäytymisen mukaisesti. Valvontakoneisto on liitetty palveluiden väliseen kommunikointikanavaan, jossa sillä on pääsy palve-

luiden väliseen tietoliikenteeseen. Valvontakoneisto valvoo palveluita liiketoimintaverkoston verkostomallin roolien mukaan. Kuvaukset on jaettu *tehtäviin*, jotka ryhmittelevät käyttäytymiskuvauksen viestit toisiinsa liittyviksi ryhmiksi. Aina kun tehtävä valmistuu, ilmoittaa valvontakoneisto tästä *BNMA*:lle eli verkostonhallinta-agentille, joka välittää tiedon eteenpäin muiden organisaatioiden agenteille. Näin huolehditaan verkoston globaalin tilan ylläpitämisestä. Kun palvelun käyttäytyminen ei vastaa kuvausta, ilmoittaa valvontakoneisto rikkeestä oman organisaationsa agentille. Sopimusta solmittaessa valvontakoneistolle rekisteröidään agentin avulla tarpeellisten verkostomallien roolien käyttäytyminen ja merkitään samalla mitkä rooleista ovat kyseessä olevan organisaation toteuttamia.

Kuvassa 5 on esitetty monitori valvomassa palvelun toimintaa. *BNMA* konfiguroi monitorin sopimuksen solmimisen yhteydessä ennen kuin varsinainen liiketoiminta sopimuksen yhteydessä aloitetaan. Konfiguroinnin yhteydessä monitorille annetaan liiketoimintaprosessikuvaus



Kuva 5: Monitori valvoo palvelun toimintaa.

ja valvontasäännöt valvonnan perusteeksi. Tämän jälkeen kun sovellus lähettää viestin toiselle sovellukselle, monitori sieppaa viestin ja tarkastaa, että sen lähettäminen on sopimuksen mukaista. Jos viesti on palvelulle sovitun käyttäytymisen mukaista, eikä riko paikallisen organisaation politiikkoja, viestin annetaan jatkaa matkaa ulos organisaatiosta. Jos taas viesti ei ole käyttäytymisen mukaista tai rikkoo organisaation politiikkoja monitori raportoi virheestä BNMA:lle ja estää viestin lähettämisen ulos organisaatiosta. Viestin vastaanottavan organisaation monitori tekee vastaavat tarkastukset ennen kuin viesti päästetään eteenpäin vastaanottavalle sovellukselle.

Riippuen monitorin konfiguraatiosta se raportoi virheistä eri tavalla BNMA:lle. Monitorilla on kolme eri tasoa raportointiin. Nämä ovat *proaktiivinen*, *aktiivinen* ja *passiivinen*. Proaktiivinen monitori aktiivisesti estää viestin kulun organisaatiosta ulos virhetilanteiden aikana ja raportoi ongelmista eteenpäin BNMA:lle. Aktiivinen monitori ei estä virheellisen viestin lähetystä, mutta raportoi aktiivisesti BNMA:lle virhetilanteista. Passiivinen monitori ei raportoi virheistä eteenpäin eikä pysäytä viestiä virhetilanteissa. Se vain rekisteröi viestin lähetyksen ja tapahtuiko virhettä vai ei. Passiivisen monitoroinnin yhteydessä toiminta voidaan jälkikäteen

tarkastaa monitorin ylläpitämästä lokikirjasta.

7 Yhteenveto

Tässä artikkelissa esiteltiin web-Pilarcos-väliohjelmiston käsitteistöä sekä tärkeimpien infrastruktuuripalveluiden yleiset periaatteet. Web-Pilarcos on nykyaikaisten liiketoimintaverkoston haasteiden ja vaatimusten pohjalta suunniteltu väliohjelmisto, joka piilottaa liiketoimintapalveluiden yhteentoimivuuteen ja liiketoimintaverkoston luomiseen ja hallintaan liittyvät ongelmat varsinaisilta palvelutoteutuksilta.

Väliohjelmisto hyödyntää palvelu- ja verkko-arkkitehtuurien periaatteita ja käyttää Web-palvelut -teknologiaa [18] kommunikointikerroksena. Web-Pilarcos -projektin lähestymistapa on käsitteistöltään ja infrastruktuuripalveluiltaan hyvin läheistä sukua sekä ODP-viitemallin [4] että viime vuosina tunnetuksi tulleen SOC-paradigman (Service-Oriented Computing) [10, 12] kanssa.

Käytännöllisen osuuden lisäksi web-Pilarcos -projektin tutkimuksen piiriin kuuluu myös liiketoimintaverkoston mallintamiseen liittyvä käsitteellinen tutkimus. Erityisesti palveluiden välisen yhteentoimivuuden takaaminen ja uudelleenkäytettävyyden tukeminen aiheutta-

vat jo käsitteellisellä ja teoreettisella tasolla ongelmia, joita ei nykyisissä liiketoimintaverkostojen tukiympäristöissä oteta huomioon. Esimerkiksi yksittäisten palveluiden yhdistäminen toiminnallisiksi liiketoimintaverkoston rooleiksi on ongelma, jonka ratkaisemiseksi täytyy käyttää hyväksi teoreettisia kehyksiä sekä prosessialgebran, tyyppijärjestelmien että kordinointimallien alueilta. Jatkossa tulemme keskittymään erityisesti liiketoimintaverkostojen dynaamisten elementtien sekä luottamuksen mallinnukseen ja hallintaan. Myös liiketoimintaverkostomallien verifiointiin liittyvät haasteet sekä infrastruktuurin jatkokehitys ja täydentäminen ovat tärkeä osa lähitulevaisuuden tutkimustoimintaamme.

Viitteet

- [1] T. Berners-Lee, R. Fielding, and L. Masinter. RFC 3986: Uniform Resource Identifiers (URI): Generic Syntax, 2005. Ladattavissa osoitteesta <http://www.ietf.org/rfc/rfc3986.txt> (30.11.2005).
- [2] ebXML Technical Architecture Project Team. ebXML Technical Architecture Specification v1.0.4. Tekninen raportti, OASIS, 2001. Ladattavissa osoitteesta <http://www.ebxml.org/specs/ebTA.pdf> (31.11.2005).
- [3] D. S. Frankel. *Model Driven Architecture - Applying MDA to Enterprise Computing*. OMG Press, 2003.
- [4] ISO/IEC JTC1/SC7. *ISO/IEC 10746: Information technology – Open Distributed Processing – Reference model*, 1998.
- [5] L. Kutvonen, T. Ruokolainen, J. Metso, and J. Haataja. Interoperability middleware for federated enterprise applications in web-Pilarcos. Teoksessa *Interoperability of Enterprise Software and Applications*. Springer-Verlag, 2005.
- [6] J. Lee, K. Siau, and S. Hong. Enterprise integration with ERP and EAI. *Commun. ACM*, 46(2):54–60, 2003.
- [7] P. Mockapetris. RFC 1035: Domain Names – Implementation and Specification, 1987. Ladattavissa osoitteesta <http://www.ietf.org/rfc/rfc1035.txt> (30.11.2005).
- [8] S. Neal, J. Cole, P. Linington, Z. Milosevic, S. Gibson, and S. Kulkarni. Identifying requirements for business contract language: a monitoring perspective. Teoksessa *Proceedings of the seventh International Enterprise Distributed Object Computing Conference*, pages 50–61. IEEE Communications, 2003. Ladattavissa osoitteesta <http://www.cs.kent.ac.uk/pubs/2003/1807> (30.11.2005).
- [9] Object Management Group (OMG). *Common Object Request Broker Architecture (CORBA): Core Specification v3.0.3*, 2004. OMG Document formal/04-03-12. Ladattavissa osoitteesta <http://www.omg.org/cgi-bin/doc?formal/04-03-12> (31.11.2005).
- [10] M. P. Papazoglou and D. Georgakopoulos. Section on Service-oriented Computing. *Communications of the ACM*, 46(10), 2003.
- [11] RosettaNet Consortium. *RosettaNet Implementation Framework: Core Specification V02.00.00*, 2004. Ladattavissa osoitteesta <http://www.rosettanet.org/> (30.11.2005).
- [12] M. P. Singh and M. N. Huhns. *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons, Ltd., 2005.
- [13] Sun Microsystems Inc. *Java2 Enterprise Edition Specification*, 2005. Ladattavissa osoitteesta <http://java.sun.com/j2ee/> (30.11.2005).
- [14] W3C. *XSL Transformations*, 1999. Ladattavissa osoitteesta <http://www.w3.org/TR/xslt> (30.11.2005).

- [15] W3C. *OWL Web Ontology Language Guide*, 2004. W3C Recommendation 10 February 2004. Ladattavissa osoitteesta <http://www.w3.org/TR/owl-guide/> (30.11.2005).
- [16] W3C. *XML Schema Documentation; Part 1: Structures, Part 2: Datatypes*, 2nd edition, 2004. Ladattavissa osoitteesta <http://www.w3.org/XML/Schema> (30.11.2005).
- [17] W3C. *Extensible Markup Language (XML)*, 2005. Ladattavissa osoitteesta <http://www.w3.org/XML/> (30.11.2005).
- [18] W3C. *W3C Web Services Activity*, 2005. Ladattavissa osoitteesta <http://www.w3.org/2002/ws/> (30.11.2005).