

Ontologiat ja tietojärjestelmien yhteistoiminnallisuus

Jari Rajanen

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaaja: Pirkko Nykänen
Toukokuu 2008

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Tekijän Nimi: Jari Rajanen
Pro gradu -tutkielma, 50 sivua
Toukokuu 2008

Yrityksillä on käytössä useita heterogeenisiä tietojärjestelmiä, jotka toimiakseen edellyttävät yhteyttä toisiin järjestelmiin. Samaan aikaan kehitetään uusia tietojärjestelmiä, jotka ainakin osaltaan perustuvat olemassa olevien sovellusten hyödyntämiseen. Tietojärjestelmiä kehitettäessä tehdään paljon työtä järjestelmien käyttämän informaation yhtensovittamiseksi. Käytettäessä käsitteitä integrointi ja yhteistoiminnallisuus (interoperability) tarkastellaan tietojärjestelmiä eri tasoilla. Tietojärjestelmiä integroitaessa sovitetaan tietojärjestelmiä yhteen teknisellä ja datan tasolla. Yhteistoiminnallisuudessa tietojärjestelmät kykenevät yhteistoimintaan niiden pystyessä hyödyntämään järjestelmien välillä siirrettävän tiedon ja tietämyksen sisältöä.

XML-kielestä on muodostunut standardi väline tiedon varastointiin ja välittämiseen tietojärjestelmien välillä. XML-kieltä käytettäessä määritellään kuitenkin ainoastaan informaation syntaktinen rakenne. Informaatioon liittyvään semanttisen tiedon esittämiseksi on kehitetty XML-kieleen perustuvia kuvauskieliä, joilla voidaan määritellä dokumenttien semanttinen sisältö.

Ontologioita on alettu hyödyntämään tietojenkäsittelyn alalla. Yleisesti hyväksytty käsitys on, että ontologioita voidaan hyödyntää tietojärjestelmien yhteistoiminnallisuudessa. Yhteistoiminnallisuuden kautta tietokoneet pystyvät ymmärtämään välitetyn tiedon semanttisen sisällön.

Palveluarkkitehtuuriin perustuvaa Web services -teknologiaa hyödynnetään yritysten tietojärjestelmien integroinnissa ja teknologiaa käytetään sekä organisaation sisäisten että organisaatioiden välisten tietojärjestelmien integrointiin. Viisiossa semanttisesta webistä älykkäät agentit suorittavat liiketoimintatapahtumia yritysten välillä. Lisäämällä semanttinen kuvaus perinteisiin Web services -sovelluksiin uskotaan pystyttävän automatisoituun palvelujen hyödyntämiseen. Jos liiketoimintaprosessien suorittaminen on agenttien vastuulla, niin samoin tulisi myös prosessien seuranta ja analysointi olla automatisoitua.

Avainsanat ja -sanonnat: yhteistoiminnallisuus, ontologia, semantiikka, tietojärjestelmä, RDF, OWL.

Sisällys

1.	Johdanto	1
1.1.	Tutkimusongelma	3
1.2.	Tutkimusmetodi ja tutkimuksen rajaus	3
2.	Ontologiat	5
2.1.	Mikä on ontologia?	5
2.2.	Muodollinen ontologia	6
2.3.	Ontologioiden muodostaminen	7
2.4.	Vaikeudet ontologioiden muodostamisessa	9
2.5.	Ontologioiden hyödyntäminen	10
2.6.	Kuinka ontologiat voivat auttaa tietojärjestelmien yhteistoiminnallisuutta?	10
2.7.	Ontologia-arkkitehtuurit	14
3.	Kuvaukielet ontologioiden esittämiseen	16
3.1.	Extensible Markup Language (XML)	16
3.2.	Resource Description Framework (RDF)	17
3.3.	Web Ontology Language (OWL)	20
4.	Yrityksen tietojärjestelmien yhteistoiminnallisuus	22
4.1.	Tietojärjestelmien yhteistoiminnallisuuden tasot	22
4.1.1.	Syntaktinen ja rakenteellinen yhteistoiminnallisuus	23
4.1.2.	Semanttinen integrointi	24
4.2.	Ontologiat ja tietojärjestelmien yhteistoiminnallisuus	24
5.	Ontologiat ja heterogeenisten tietokantojen yhteistoiminnallisuus	26
5.1.	XML kaaviot	27
5.2.	Ontologioiden käyttö tietokantojen suunnittelussa	28
5.3.	Ontologioiden hyödyntäminen käytön aikana	28
5.4.	Ontologioiden käyttö tietämyksenhallinnassa	28
6.	Tietojärjestelmäintegraatiosta tietojärjestelmien yhteistoiminnallisuuteen	30
6.1.	Väliohjelmistoista Web service –teknologian käyttöön tietojärjestelmäintegraatiossa	31
6.2.	Web services –teknologian käyttö tietojärjestelmäintegraatioon	32
6.2.1.	Web services arkkitehtuuri	32
6.2.2.	Web services -teknologian käyttö	33
6.3.	Semanttinen Web service –teknologia ja tietojärjestelmien yhteistoiminnallisuus	34
6.3.1.	WSDL ja ontologiat	35

6.3.2. Palveluiden kuvaus ontologioita käyttäen Web services -teknologiassa	35
7. Liiketoimintaprosessien yhteistoiminnallisuus ontologioiden avulla	37
7.1. Semantiikka elektronisen kaupan järjestelmissä	38
7.2. Workflow-mallin luominen liiketoimintasopimuksesta sähköisessä sopimisessa ontologian avulla	40
7.3. Liiketoimintaprosessin monitorointi	41
8. Yhteenveto	43
Viitteet :	47

1. Johdanto

Yrityksillä on käytössä useita itsenäisiä tietojärjestelmiä, joita on kehitetty erilaisiin tarpeisiin ja eri konteksteihin. Samaan aikaan on olemassa tarve kehittää uusia, usein yhä monimutkaisempia tietojärjestelmiä, jotka ainakin osaltaan perustuvat olemassa olevien sovellusten hyödyntämiseen. Järjestelmien käyttämän informaation integroimiseksi on yrityksissä tehty paljon töitä, jonka ansiosta eri ohjelmistojen on mahdollista jakaa käyttämäänsä dataa keskenään. XML-kielestä on muodostunut standardi väline jakaa dataa ohjelmistojen välillä. XML-kielen suosio perustuu sen soveltuvuuteen rakenteisen datan esittämiseen. XML-kieltä on jo pitkään hyödynnetty järjestelmien yhteisenä kielenä informaation siirtoon heterogeenisten tietojärjestelmien välillä. Vaikka XML:n avulla voidaan ratkaista syntaktinen yhteensopivuus tietojärjestelmien välillä, tarvitaan usein ihmisen ymmärtämystä päättelemään esimerkiksi käytetäänkö kahta tietokannan kenttää samassa semanttisessa merkityksessä.

Käsitteet tietojärjestelmien integrointi ja yhteistoiminnallisuus (interoperability) ovat lähellä toisiaan ja monessa yhteydessä niitä käytetään osin päällekkäin ja jopa toistensa synonyymeina. Tietojärjestelmien toimintaa tarkasteltaessa käytetään käsitteitä integrointi ja yhteistoiminnallisuus tarkastelun eri tasoilla ja hieman eri näkökulmasta tietojärjestelmiä tarkasteltaessa. Tietojärjestelmien integroinnilla tarkoitetaan järjestelmien kykyä välittää ja vaihtaa dataa, viestejä ja tapahtumia keskenään tai tiiviin liitoksen luomista järjestelmien prosessien välillä [Pollock, 2001]. Tietojärjestelmien yhteistoiminnallisuudella tarkoitetaan tietojärjestelmien kykyä vaihtaa informaatiota ja hyödyntää vaihdettua informaatiota [Geraci *et al.*, 1991]. Integroinnilla tarkoitetaan siis tietojärjestelmien yhteensovittamista teknisellä ja datan tasolla. Yhteistoiminnallisuudella tarkoitetaan tietojärjestelmien kykyä yhteistoimintaan niiden pystyen hyödyntämään järjestelmien välillä siirrettävän tiedon ja tietämyksen sisältöä. Tietojärjestelmien yhteistoiminnallisuuden mahdollistamiseksi on välttämätön edellytys, että yhteistoiminnassa olevat järjestelmät kykenevät kommunikoimaan teknisellä tasolla.

Eri järjestelmillä ja niitä kehittäville ja käytävillä ihmisillä on erilaisia näkökulmia sovellusalueilta ja myös siitä, mikä on oleellista asioita käsiteltäessä ja esitettäessä. Eri ohjelmistoissa käytetään usein eri termistöjä ja termejä käytetään eriävissä semanttisissa merkityksissä. Tämä vaikeuttaa heterogeenisten tietojärjestelmien yhteensovittamista, koska sujuvan ja yhtenäisen informaation välittäminen edellyttää informaation semanttisen merkityksen huomioonottamista järjestelmiä yhteen sovitettaessa.

Tietojenkäsittelyn alalla ontologian avulla voidaan mallintaa johonkin alaan liittyvä tietämys määrittelemällä käytettävä sanasto ja sen rakenne kuvaamalla termien väliset suhteet. Ontologioiden järjestelmät pystyvät ymmärtämään tiedon aiotun merkityksen esittäessä ontologiassa entiteettien väliset sääntölogisten sääntöjen avulla. Ontologioiden esittämiseksi tietojärjestelmissä on kehitetty XML-kieleen perustuvia kuvauskieliä. Yleisesti on hyväksytty käsitys, että ontologioita voidaan hyödyntää pyrittäessä tietojärjestelmien yhteistoiminnallisuuteen. Ontologioita luodaan käyttötarkoituksesta riippuen erilaisilla erikoistamisen tasoilla ja eri sovellusaloille. Pyrkimällä ontologioiden uusiokäyttöön tavoitellaan kustannussäästöjä ontologioiden muodostamiseen tarvittavan työmäärän pienentyessä. Lisäksi käytettäessä ontologioita mahdollistamaan tietojärjestelmien yhteistoiminnallisuutta pyritään luomaan ontologiat siten, että ne olisivat vertailukelpoisia keskenään.

Ontologioita käytetään tietojärjestelmissä termien semanttisen merkityksen kuvaamiseen, minkä avulla tietokoneet pystyvät päättämään, käyttävätkö kaksi eri järjestelmää käsitettä samassa semanttisessa merkityksessä. Edellytyksenä ontologioiden jakamiselle heterogeenisten tietojärjestelmien välillä on, että järjestelmät käyttävät samaa kieltä ontologioiden kuvaamiseen. Ontologioiden kuvaamiseen on kehitetty XML-kieleen perustuvia kuvauskieliä, kuten esimerkiksi Web Ontology Language (OWL).

XML-kielen käyttö on levinnyt laajalle tiedon välityksessä itsenäisten tietokantajärjestelmien välillä. Yleensä tarvitaan kuitenkin ihmisen ymmärtämystä päättämään tietokannan kenttien sisällön merkitys. Hajautetuissa tietojärjestelmissä on mahdollista suorittaa tietokantojen yhteensovittaminen standardoitua XML kaaviota käyttäen. Hyödyntämällä käytön aikana järjestelmien jakamaa ontologiaa XML kaavion määrittelyssä voidaan varmistaa yhteistoiminnassa olevien tietojärjestelmien käyttävän tietokannan elementtejä samassa käsitteellisessä merkityksessä ja siten on kenties mahdollista automatisoida tietokantojen yhteensovittamista.

Web services -teknologiaa hyödynnetään yritysten tietojärjestelmissä tavoitteena mahdollistaa tietojärjestelmien integrointi Web services -teknologian soveltuessa käytettäväksi sekä organisaation sisäisesti että organisaatioiden välillä. Perinteisten Web services -järjestelmien sujuvaa yhteistoiminnallisuutta vaikeuttaa palveluiden ja toiminnallisuuden kuvaukseen liittyvän semantiikan muodollisen kuvauksen puuttuminen palvelun kuvauksesta. Luomalla semanttisia Web services -palveluita uskotaan pystyvän luomaan tie kohti (puoli)automaattista yritysten tietojärjestelmien integrointia ja dynaamista uusien sovellusten laatimista.

Visiossa semanttisesta webistä [Berners-Lee *et al.*, 2001] suoritetaan liiketoimintatapahtumia yritysten välillä älykkäiden agenttien toimesta. Visiossa älykkäät agentit hyödyntävät yhteistoimintaan kykeneviä tietojärjestelmiä liiketoimintaprosessien suorittamisessa. Yrityksen toiminnalle liiketoimintaprosessien etenemisen seuraaminen voi olla elintärkeää. Jos prosessien läpivienti tapahtuu automatisoidusti eri järjestelmien välillä, niin tällöin tulisi myös prosessien seuranta ja analysointi pystyä automatisoimaan yrityksen toiminnan rationalisoimiseksi.

1.1. Tutkimusongelma

Tämän opinnäytetyön tarkoituksena on lisätä ymmärtämystä ontologioiden käytöstä tietojärjestelmien yhteistoiminnallisuuteen analysoimalla aiheeseen liittyviä tieteellisiä artikkeleita ja muodostamalla sen perusteella yhtenäinen kuva aiheesta. Tutkimusongelmana on: Miten ontologioita voidaan hyödyntää tietojärjestelmien yhteistoiminnallisuuteen? Tutkimusongelmaan liittyen ja sen selkeyttämiseksi lisäksi ensiksi selvitetään mitä tarkoitetaan ontologialla tietojärjestelmien yhteydessä sekä mitä tarkoitetaan tietojärjestelmien yhteistoiminnallisuudella ja mitä hyötyä siitä voi olla.

1.2. Tutkimusmetodi ja tutkimuksen rajaus

Hevner *et al.*:n [2004] esittämän mallin mukaan tietojärjestelmien tutkimuksessa osoitetaan oikeaksi/ evaluoidaan alan tietämykseen perustuvia teorioita ja luotuja artefakteja eri tutkimustapojen avulla, joita ovat:

- Analyyttinen tutkimus
- Case study
- Kokeellinen tutkimus
- Kenttätutkimus
- Simulointi

Tietojärjestelmien tutkimuksen tuloksia voidaan arvioida liiketoiminnan tarpeiden tyydyttämisen perusteella, kun tutkimustuloksia sovelletaan relevantissa ympäristössä [Hevner *et al.*, 2004]. Hevner *et al.*:n mukaan tietojärjestelmien tutkimus koostuu käyttäytymistieteellisestä ja suunnittelutieteellisestä tutkimuksesta. Tässä opinnäytetyössä on käytetty käsitteellisteoreettista tutkimusotetta analysoiden materiaalina käytettyä alan tieteellistä kirjallisuutta. Tämä tutkimus on siis tietojärjestelmätieteen suunnittelutieteellistä tutkimusta, jossa analysoidaan olemassa olevia käsitteitä kirjallisuuslähteiden perusteella ja pyritään tuottamaan käsitys siitä, miten ontologioita voidaan hyödyntää tietojärjestelmien yhteistoiminnallisuuden toteuttamisessa.

Yhteistoiminnallisuuden käsitettä voidaan soveltaa useiden erityyppisten informaatiotekniikan sovellusten yhteydessä. Tässä työssä rajoitutaan tarkastelemaan ontologioiden käyttöä tietojärjestelmien yhteistoiminnallisuuteen yritysten tietojärjestelmien välillä.

2. Ontologiat

2.1. Mikä on ontologia?

Termi ontologia tulee alkuaan filosofian tutkimusalalta, jossa se tarkoittaa opia olevaisuudesta. Filosofiansa ontologian tavoitteena on luoda mahdollisimman yksiselitteinen ja tyhjentävä kuvaus reaali maailmasta tai sen osasta kuvaten kaikki siihen kuuluvat entiteetit ja niiden väliset suhteet [Smith, 2003]. Terminä ontologia on syntynyt 1600-luvulla, mutta jo Aristoteles tutki ontologiaa käsitteenä. Käsitteenä ontologia on levinnyt myös muille tieteenaloille ja tästä johtuen on termi ontologia saanut myös useita eri merkityksiä. Guarino [1995b] pyrkii selventämään käsitteitä käyttämällä tieteenalasta termiä suurella alkukirjaimella kirjoitettuna (englanniksi Ontology) ja kirjoittamalla termi pienellä alkukirjaimella (englanniksi ontology), kun termillä tarkoitetaan reaali maailman kuvausta.

Käsitteistöön liittyvänä ongelmana Guarino [1998] tuo esiin myös sen, että toisinaan on epäselvää, tarkoitetaanko termillä ontologia filosofista teoriaa, jonka sisältö on sama esityskielestä riippumatta vai tarkoitetaanko ontologialla teorian konkreettista esitystä, joka on esityskielestä riippuvainen. Guarino päätyy tässä yhteydessä käyttämään filosofisesta teoriasta termiä käsitteellistämisen (conceptualization) ja konkreettisesta teorian esittämisestä termiä ontologia. Tällä perusteella samasta käsitteistöistä voidaan muodostaa useita eri ontologioita. Näin ollen ontologia on artefakti, joka on mahdollista lukea, myydä tai muuten jakaa fyysisesti [Guarino, 1995b].

Ontologia voidaan nähdä olevan kohdemaailman organisaation ja sen luonteen kuvaus, joka on riippumaton tietämyksen luonteesta mikä meillä on kohdemaailmasta [Guarino, 1995a]. Gruber [1993,1995] määrittelee ontologian olevan käsitteellistämisen eksplisiittistä erikoistamista. Gruberin [2003] mukaan ontologia on käsitteiden ja niiden välisten suhteiden määritelmä. Guarino [1998] tarkentaa edellä mainittua Gruberin määritelmää määrittelemällä ontologian olevan looginen teoria, jolla määritellään muodollisen sanaston aiottu merkitys. Gruber [2008] selkeyttää yllä mainittua määritelmäänsä, jossa todetaan ontologian olevan käsitteellistämisen erikoistamista, sanomalla ontologian määrittelemiseksi olevan välttämätöntä, että ontologiassa määritellään käsitteet, niiden väliset suhteet ja muut erittelevät piirteet jotka ovat relevantteja sovellusalan mallintamiseen. Gruberin mukaan ontologian esitysmuotona on rakenteellinen sanasto, joka koostuu luokista, niiden välisistä suhteista ja ominaisuuksista, mikä tarjoaa sanaston tarkoituksen ja muodolliset säännöt sen

johdonmukaiseen käyttöön. Sanasto koostuu yleensä kohdemaailman käsitteistöä ja näiden käsitteiden väliset suhteet määritellään hierarkkisesti. Käsitteiden väliset suhteet määritellään yleensä käyttäen ensimmäisen kertaluvun logiikkaa, jolloin logiikan sääntöjen avulla pystytään ilmaisemaan myös sanaston aiottu semanttinen merkitys [Guarino, 1998].

Gruberin [2008] mukaan tietojenkäsittelyn ja informaatiotieteiden piirissä ontologian voidaan määritellä käsittävän primitiivit, joiden avulla voidaan mallintaa johonkin alaan liittyvä tietämys tai diskurssi. Hänen mukaan ontologian määrittelevät primitiivit muodostuvat yleensä luokista (joukoista), attribuuteista (tai ominaisuuksista) ja suhteista (tai luokkien jäsenten välisistä suhteista). Näiden edellä mainittujen primitiivien määritelmät sisältävät myös primitiivien aiottu merkityksen ja määrittelyalan, jolla primitiivit muodostavat loogisesti yhtenäisen kokonaisuuden.

2.2. Muodollinen ontologia

Formaali ontologia (formal ontology) on täsmällinen, selkeä ja looginen kuvaus kohdemaailmasta [Guarino, 1997]. Tietojärjestelmien yhteydessä termin *muodollinen* käyttö viittaa yleensä siihen, että ontologia on ilmaistu tietokoneiden tulkittavissa olevassa muodossa. Tällöin ontologiassa määriteltyjen käsitteiden ja niiden ilmentymien välisten suhteiden kuvaus on useimmiten määritelty muodollisesti käyttäen ensimmäisen kertaluvun predikaattilogiikkaa, joka tarjoaa päättelysäännöt tietokoneiden käytettäväksi. Guarino [1998] kuitenkin käyttää termiä formaali ontologia pääasiassa kuvaamaan filosofian tutkimus-alaa. Guarinon [1995a] mukaan formaalissa ontologiassa ei olla niinkään kiinnostuneita joidenkin tiettyjen yksilöiden olemassaolosta vaan pikemminkin kuvataan niiden olemassaolon olomuotoa. Hänen mukaan formaalilla ontologialla tarkoitetaan teoriaa, jolla kuvataan kuvattavan maailman entiteettien eritteleviä ominaisuuksia (olioita, tapahtumia, alueita, aineksen määrää...) tai kuvataan kuvattavan maailman mallintamiseen käytettävien metatason luokkia (käsitteitä, ominaisuuksia, määriä, tiloja (state), rooleja...).

Formaalit ontologiat koostuvat yleensä termien määrittelyistä. Käytännössä formaali ontologia tarjoaa käytettäväksi sanaston sovellusalueen kuvaamiseksi sekä määrittelee käytettävän sanaston rakenteen kuvaamalla sovellusalueen kuvauksessa käytettyjen termien väliset suhteet. Formaaleihin ontologioihin sisältyy yleensä käsitteet attribuutteineen, käsitteiden väliset suhteet ja rajoitukset sekä oliot, jotka ovat käsitteiden esiintymiä [Gali *et al.*, 2004].

2.3. Ontologioiden muodostaminen

Ontologiaa muodostettaessa tarkoituksena on määritellä kuvattava kohdealue siten, että termien tarkoitus on mahdollisimman selkeä. Ontologia pyritään määrittelemään mahdollisimman eksplisiittisesti niin tarkalla abstraktiotasolla kuin mahdollista. Toisaalta samaan pyritään samaan aikaan säilyttämään riittävän korkea abstraktiotaso, jotta ontologiaa voidaan soveltaa riittävän laajalti [Uschold and Gruninger, 1996]. Määriteltävä ontologia on myös riippumaton tiedonmallinnuksen strategiasta tai toteutuksesta [Smith, 1995].

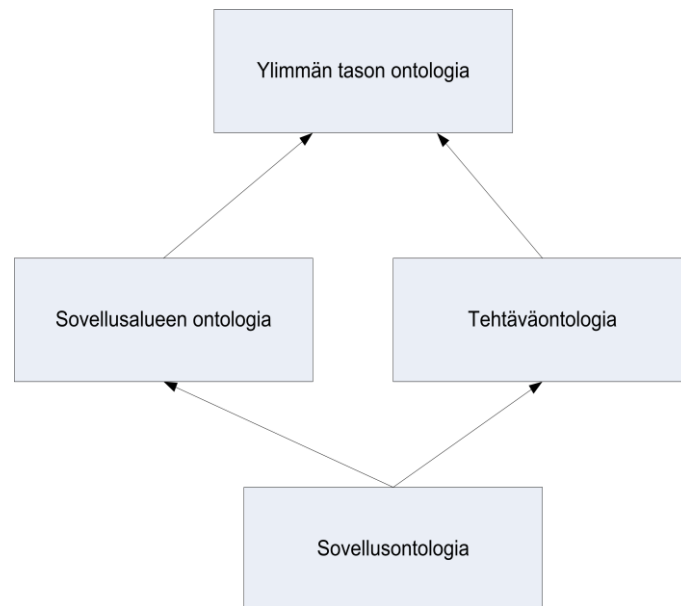
Muodollisesti ontologia voi olla useissa eri muodoissa, mutta siitä huolimatta se sisältää ainakin käytettävän sanaston ja myös jonkin määrittelyn sanaston merkityksestä [Uschold and Gruninger, 1996]. Gruninger ja Uschold luokittelevat ontologiat muodollisuutensa perusteella eri luokkiin:

- Erittäin vapaamuotoinen (highly informal): luonnollisella kielellä määritelty ontologia.
- Puoli-vapaamuotoinen (semi informal): rakenteisella tai rajoitetulla luonnollisella kielellä määritelty ontologia.
- Muodollinen (formal): muodollisesti määritellyllä kielellä, kuten esimerkiksi UML, määritelty ontologia.
- Täsmällisesti muodollinen (rigorously formal): muodolliseen logiikkaan perustuvalla kielellä määritelty ontologia.

Luonnollisella kielellä vapaamuotoisesti määritellyt ontologiat voivat soveltua ihmisten käytettäväksi. Mutta haluttaessa tietokoneiden pystyvän hyödyntämään ontologiaa, täytyy ontologian sisältää muodolliset päättelysäännöt joiden avulla tietokoneen on mahdollista päätellä termien väliset suhteet. Analysoimalla loogisia päättelysääntöjä, termien ominaisuuksia ja niiden välisiä suhteita sekä rajoituksia voivat tietokoneet päätellä tiedon aiotun merkityksen ihmisten käsityskykyä vastaavasti.

Ontologioita voidaan jakaa eri luokkiin myös niiden käyttötarkoituksen ja abstraktiotason mukaan. Luotaessa ontologioita yksi vaihtoehto on määritellä ne hyvin yleisellä tasolla, jolloin ne sopivat käytettäväksi usealla eri alalla. Vastaavasti voidaan luoda myös hyvin alaspesifisiä ontologioita. Koska ontologioita voidaan luoda hyvin erilaisiin käyttötarkoituksiin ja erilaisilta aloilta, niin ontologiat kuvaavat kohdealuettaan hyvin erilaisilla tarkkuuksilla. Yleinen ontologia on yleensä liian geneerinen, jotta sitä voitaisiin sinällään käyttää kohdealueen kuvaukseen. Sen sijaan erikoistamalla yleistä ontologiaa voidaan saavuttaa riittävä tarkkuus kohdealueen kuvaamiseksi. Guarino [1997] jakaa ontologiat eri tyyppeihin sen mukaan, kuinka erikoistettuja kuvauksia kohde maailmasta ontologiat ovat. Kuvassa 1 esitetään, kuinka sovellusalueen ontologia ja

tehtäväkohtainen ontologia on muodostettu erikoistamalla ylimmän tason ontologiaa. Ylimmän tason ontologiat kuvaavat maailmaa hyvin yleisellä tasolla, määrittelemällä esimerkiksi termit kuten olio, aika, asia tai tila. Sovellusalueen ontologia määrittelee sanaston yleisellä tasolla jollekin alalle, kuten esimerkiksi lääketiede tai auto. Vastaavasti tehtäväontologia määrittelee sanaston yleisellä tasolla jollekin tehtävälle, kuten esimerkiksi myyminen tai tutkimus, erikoistamalla ylimmän tason ontologiaa. Jossain tietyssä sovelluksessa käytettävä ontologia muodostetaan usein erikoistamalla sekä sovellusalueen ontologiaa että tehtäväontologiaa.



Kuva 1 Ontologiatyypit Guarinon [1997] mukaan

Ontologiaa luotaessa siinä kuvataan asioita, jotka esiintyvät kohdealueella ja voidaan ajatella, etteivät kohdealueella esiintyvät asiat ole riippuvaisia siitä mihin tarkoitukseen ontologiaan ollaan tekemässä. Käytännössä ontologiat luodaan kuitenkin yleensä johonkin tiettyyn käyttötarkoitukseen ja tällöin ontologiassa kuvataan ainoastaan termit, jotka ovat relevantteja ontologian ajattelun käyttöalueen tehtäviä ajatellen. Käytännössä onkin todennäköistä, ettei ontologia kata kaikkia mahdollisia kuviteltavissa olevia sovelluskohteita [Chandrasekaran *et al.*, 1999]. Tästä johtuen yksittäinen ontologia soveltuu useimmiten paremmin käytettäväksi tehtävään, jota silmälläpitäen se on laadittu tai tehtävään, joka on lähellä alkuperäistä tehtävää kuin tehtävään, joka suuresti poikkeaa alkuperäisestä käyttötarkoituksesta.

Ontologian muodostaminen yhdistämällä useita eri tasoilla laadittuja ontologioita säästää ontologian laatimiseen tarvittavaa työtä aiemmin laadittujen ontologioiden uudelleen käyttöön perustuen. Yhdistelemällä samaan ylemmän

tason ontologiaan perustuvia ontologioita saadaan lisäksi hyötynä yhdistelmällä luotujen ontologioiden olevan vertailukelpoisia keskenään, koska näin muodostettavissa ontologioissa käytetään yhteistä sanastoa ja ontologiat jakavat saman käsitteellistämisen. Ontologioiden ollessa vertailukelpoisia keskenään se helpottaa niitä hyödyntävien järjestelmien yhteensovittamista ja tukee siis yhteentoimivuutta.

2.4. Vaikeudet ontologioiden muodostamisessa

Käytettäessä ontologioita tietojärjestelmien yhteistoiminnallisuuteen ideaalita-paus ontologiasta olisi kaiken kattava neutraali ontologia. Käytännössä on kuitenkin hylätty ajatus kaiken kattavan ontologian muodostamisesta ajatuksen ollessa epärealistinen. Määriteltäessä ontologiaa sen tulisi olla yleinen ja neutraali kuvaus kohdealueesta ontologian jakavien yhteisöjen kesken; samalla sen tulisi olla mahdollisimman laaja-alainen sekä luoda tarkka ja yksikäsitteinen kuvaus kohdealueesta [Smith, 2003]. Yksi ratkaisu yllä mainitun kaksijakoisen vaatimuksen täyttämiseksi on määritellä yleinen ylimmän tason ontologia, jota erikoistetaan tarpeen mukaan.

Ontologian avulla voidaan kuvata paitsi yksinkertaisia faktoja niin myös uskomuksia, tavoitteita, hypoteeseja ja arvioita kohdealueelta [Chandrasekaran *et al.*, 1999]. Ontologian jakavilla osapuolilla on usein eriäviä käsityksiä edellä mainitun kaltaisista kohdealueen käsitteistä. Myös tietojärjestelmiä toimittavat yritykset käyttävät mielellään omia termejään ja toteutustapoja ontologioita määriteltäessä [Ushold and Gruninger, 2004]. Edellä mainituista seikoista johtuen voi joissain tapauksissa olla vaikeaa löytää yhteisymmärrys organisaatioiden välillä siitä, kuinka sovellusalueen tietämystä kuvataan ontologiassa ja miten organisaatioiden jakama ontologia tulisi muodostaa. Eri organisaatioiden toimiessa hieman eri tavoin ne saattavat omata erilaisen kuvan kohdealueen käsitteiden rakenteesta ja käsitteistä käytettävistä termeistä. Standardoinnilla voidaan sopia käytettävistä termeistä ja niiden mahdollisesta kodifioinnista [RIDE Consortium, 2008]. Standardoinnilla onkin siten merkittävä rooli pyritäessä yleisesti hyväksytyjen ontologioiden määrittelyyn tietojärjestelmien yhteistoiminnallisuuden mahdollistamiseksi.

On lähes välttämätöntä, että ontologiat muuttuvat ajan myötä. Koska ontologiat usein laajentavat ja käyttävät uudelleen toisia ontologioita, heijastuvat ontologiaan tehdyt muutokset kaikkiin sitä käyttäviin ontologioihin sekä sovelluksiin, jotka hyödyntävät näitä ontologioita. Yksinkertaisetkin muutokset ontologiassa saattavat aiheuttaa odottamattomia seurauksia sitä uudelleen käyttävissä ontologioissa ja sovelluksissa [Noy and Klein, 2004].

2.5. Ontologioiden hyödyntäminen

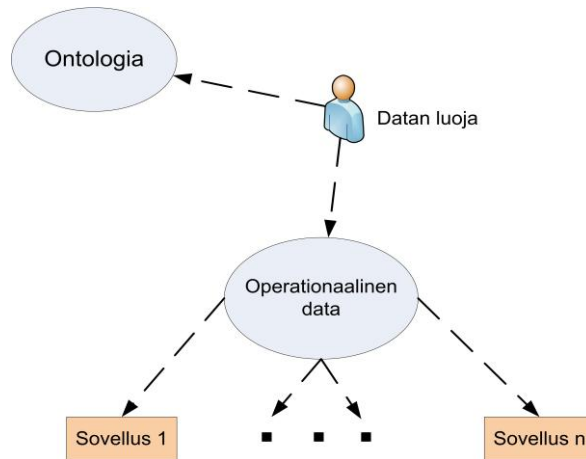
Ontologioita voidaan hyödyntää useassa tarkoituksessa. Mahdollisia käyttö-tarkoituksia voi olla esimerkiksi tietämyksen välittäminen ihmiseltä ihmiselle, ihmisen käsittämän tietämyksen välittäminen tietojärjestelmien hyödynnettäväksi tai tietämyksen välittäminen tietojärjestelmien välillä. Esimerkkinä ihmiseltä ihmiselle tapahtuvasta tietämyksen välittämisestä on ontologinen analyysi. Ontologinen analyysi selkeyttää tietämyksen rakennetta, missä analyysin avulla voidaan kerätä sanasto, joka kuvaa kohdealueen tietämystä [Chandrasekaran *et al.*, 1999]. Ontologisen analyysin tuotteena saadaan sovellusalueesta käsitteellinen malli, jota voidaan hyödyntää tietojärjestelmää toteutettaessa. Ontologiat tukevat myös tietämyksen jakamista, sillä kohdealueelta muodostettu ontologia on mahdollista antaa myös muiden sitä tarvitsevien käytettäväksi. Yleensä voidaan analyysin tuloksena saatava ontologia ilmaista siten, että ontologia on sekä ihmisten että tietokoneiden hyödynnettävissä olevassa muodossa.

Guarino [1998] sanoo tiedon integroinnin olevan merkittävä ontologioiden sovellusalue. Koska sama ontologia voidaan antaa useampien tietojärjestelmien käytettäväksi, pystytään ontologioita jakamalla jakamaan tietämystä tietojärjestelmien välillä. Jotta ontologioita voidaan hyödyntää tietojärjestelmien yhteistoiminnallisuuden saavuttamiseksi, täytyy kaikkien osapuolten käyttää yhteistä kieltä ontologioiden esittämiseen [Uschold and Gruninger, 2004]. Kahden tietojärjestelmän jakaessa saman ontologian, jossa määritellään kohdealuetta kuvaava sanasto, on tietojärjestelmillä yhtenevä kuva kohdealueesta. Käytettävän sanaston jakaminen eri tietojärjestelmien kesken käyttö tukee tietojärjestelmien integrointia.

2.6. Kuinka ontologiat voivat auttaa tietojärjestelmien yhteistoiminnallisuutta?

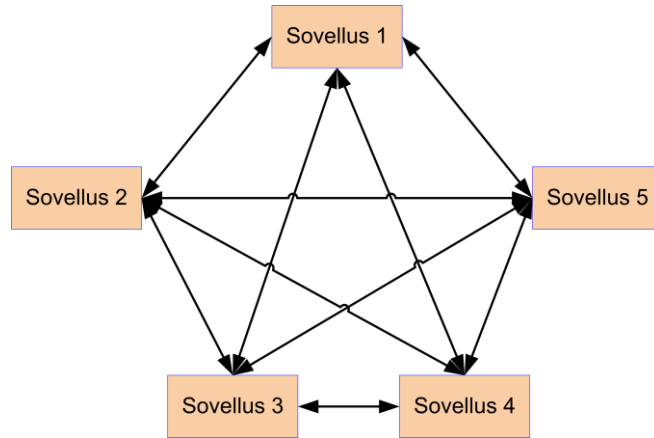
Organisaation sisällä saattaa olla käytössä useita ohjelmistoja, jotka eivät kykene yhteistoimintaan johtuen eriävien käsitteistöjen käytöstä tiedon esittämiseen. Uscholdin ja Gruningerin [2004] mukaan tällaisessa tilanteessa yritys saattaa saavuttaa suurta hyötyä soveltamalla neutraalin julkaisun (neutral authoring) lähestymistapaa järjestelmien käyttämän termistön yhdenmukaistamisessa. Kuva 2 havainnollistetaan neutraalin julkaisun mallia, jossa kehitetään koko sovellusalan kattava neutraali ontologia. Neutraalia ontologiaa hyödynnetään määrittelemään kaikkien sovellusten käyttämä data siten, että sovelluksille yhteisessä datassa käytetään neutraalin ontologian mukaista termistöä. Tarvittaessa luodaan sovelluskohtaisia kääntäjiä, jotka muuntavat datan tästä kaikille sovelluksille yhteisen tietomallin mukaisesta muodosta sovelluksen käyttöön

sopivaksi. Kaikkien sovellusten alan kattavan neutraalin ontologia luominen on vaativa prosessi, mutta hyötynä voidaan saada tietämyksen uusiokäytön mahdollistaminen, tiedon ylläpidon helpottuminen ja tietämyksen kehittyminen pitkällä aikavälillä [Uchold and Gruninger, 2004].



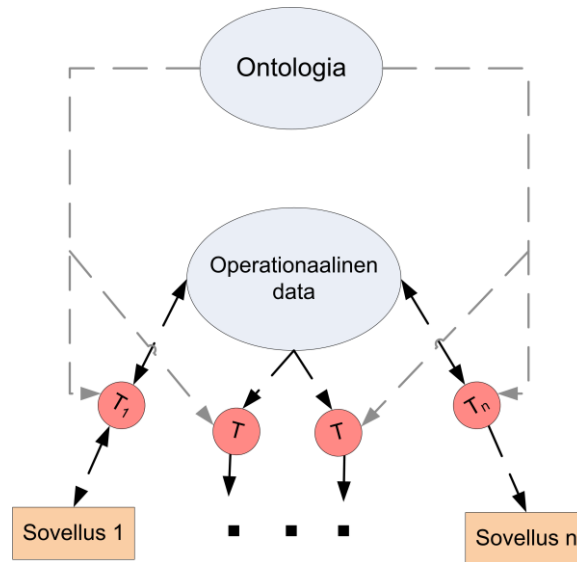
Kuva 2 Neutraali julkaisu Ucholdia ja Gruiningeria [2004] mukailleen

Yrityksissä on yleisesti käytössä useita eri järjestelmiä, jotka toimiakseen edellyttävät yhteistoiminnallisuutta. Perinteiset järjestelmät (legacy systems) ovat usein heterogeenisiä, jolloin siirrettäessä informaatiota järjestelmien välillä on tarpeen muuntaa tietoa eri muotoon ja käyttää erilaisia esitystapoja. Kuva 3 on havainnollistettu perinteisten järjestelmien yhteensovittamisen ongelmaa. Kuten Kuva 3 voidaan nähdä, johtaa sovitusten luomien kaikkien yhteistoimintaan osallistuvien järjestelmien kesken helposti hyvin suureen sovitimien määrään. Yhteistoimintaan osallistuvien järjestelmien lukumäärän kasvaessa kasvaa sovitusten lukumäärä $O(N^2)$. Sovittimien kehittämisen lisäksi on myös sovitimien ylläpito usein vaativaa ja edellyttää siten suurten resurssien käyttöä ylläpitoon.



Kuva 3 Yhteistoiminallisuuden ongelma

Pyrittäessä saavuttamaan yhteistoiminnallisuus perinteisten järjestelmien välillä joudutaan yleensä aina luomaan käännöksiä käytetyistä termeistä ja muokkaamaan tiedon esitystapaa, jotta järjestelmät pystyvät kommunikoimaan keskenään. Yleensä voidaan olettaa, ettei pystytä määrittelemään globaalia ontologiaa, joka vastaisi kaikkien yhteen sovitettavien järjestelmien käsitteistöä. Myös tiedon siirtoon eri järjestelmät käyttävät erilaisia formaatteja. Ontologiaa voidaan hyödyntää neutraalina välittäjänä olemassa olevien järjestelmien välillä, jolloin on mahdollista selvittää huomattavasti pienemmällä määrällä sovittimia [Uschold and Gruninger, 2004]. Ontologian käyttöä välittäjänä mahdollistamaan yleinen pääsy informaatioon on havainnollistettu Kuva 4. Toimintaperiaatteena neutraalin välittäjän käytössä on, että välittäjä suorittaa ensin käännöksen lähdejärjestelmästä neutraaliin muotoon, jolloin tiedon semanttinen merkitys on määritelty neutraalin ontologian mukaisesti, ja sen jälkeen välittäjä suorittaa käännöksen neutraalista muodosta kohdejärjestelmän muotoon. Neutraalin ontologian sisältäessä kaikki kohdejärjestelmien käsitteet on mahdollista suorittaa käännos siten, että siirrettävän informaation semanttinen sisältö säilyy muuttumattomana. [Uschold and Gruninger, 2004]



Kuva 4 Yleinen pääsy informaatioon Uscholdia ja Gruningeriä mukailten [2004]

Hyödyntämällä ontologiaa sovellusten välillä tapahtuvan informaation siirtämisen hoitavien sovittimien toteutuksessa on mahdollista jakaa ja siirtää tietoa sovellusten välillä, vaikka sovellukset käyttävät eriäviä termejä tai formaatteja. Näin saavutettava yhteistoiminnallisuus sovellusten välillä perustuu ajatukseen, että käytettävä ontologia on molempien osapuolten hyväksymä standardi jota hyödynnetään muunnoksessa (converting) ja sovittamisessa (mapping). [Uschold and Gruninger, 2004]

Käytännönläheinen esimerkki yllä kuvatun kaltaisen arkkitehtuurin soveltamisesta annetaan RIDE raportissa [2008]. Raportissa esitetään tilanne missä yhteistoiminnallisuutta tavoitellaan kahden itsenäisen terveydenhuollon tietojärjestelmän välillä, jotka käyttävät eri standardien mukaisia termien määrittelyjä. Järjestelmien yhteistoiminnallisuuden mahdollistamiseksi hyödynnetään ontologiaa, jossa luodaan sovitus kahden eri standardin mukaisten termien välille määrittelemällä ontologiassa eri standardien mukaisten termien vastaavuudet ontologiassa. Siirrettäessä tietoa järjestelmien välillä pystytään ontologiaa käyttämällä päättämään, milloin järjestelmien käyttämät termit tarkoittavat samaa käsitettä siten, että päättelyssä otetaan huomioon myös termien semanttinen sisältö. Standardien kehittäminen onkin avainasemassa yhteistoiminnallisuuden mahdollistamisessa. Ontologioilla on tärkeä tehtävä järjestelmien yhteistoiminnallisuudessa varmistaa, että yhteistoiminnassa olevat tietojärjestelmät käyttävät termejä samassa semanttisessa merkityksessä, kun toimitaan arkkitehtuurissa jossa pyritään luomaan yleinen pääsy informaatioon.

Kaupallisessa ympäristössä toimiviin tietojärjestelmiin tehdään muutoksia lyhyin aikavälein johtuen liiketoimintaympäristössä nopeasti tapahtuvista

muutoksista, jolloin myös tietojärjestelmille asetetut vaatimukset muuttuvat nopeassa tahdissa. Seurauksena nopeasta liiketoimintamallien muutostahdista on ontologioihin perustuvien käänösmoduuleiden ylläpidolla ollut vaikeuksia pysyä muutosten tahdissa [Smith, 2003].

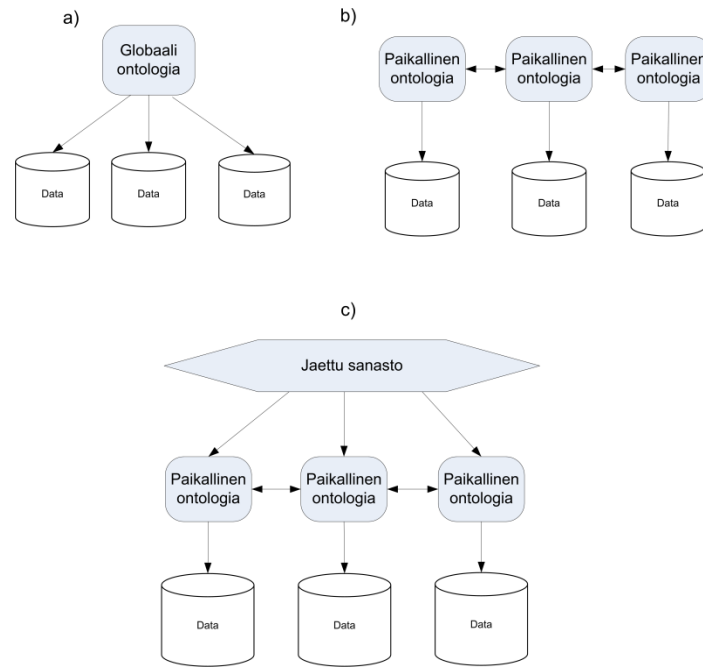
2.7. Ontologia-arkkitehtuurit

Käytettäessä ontologioita tietojärjestelmien yhteistoimintaan, voidaan löytää kolme eri mahdollista mallia kuinka ontologia(t) sijoittuvat järjestelmässä: single-ontologia malli, multiple-ontologia malli ja hybridimalli [Hübner *et al.*, 2001]. Eroavaisuudet eri mallien välillä ovat lähinnä siinä, kuinka hyvin eri järjestelmien käyttämät ontologiat ovat vertailukelpoisia. Kuva 5 esitetään eri mallien periaatteita.

Single ontologia malli, tapaus a) Kuva 5, on keskitetty malli, jossa on niemensä mukaisesti käytössä yksi kaikkien integroitavien tietojärjestelmien välillä jaettu ylimmän tason ontologia.

Multiple-ontologia malli, tapaus b) Kuva 5 on hajautettu malli, jossa jokaisella tietojärjestelmällä on oma ontologiansa. Koska useaa ontologiaa käytettäessä tietojärjestelmät eivät jaa yhteistä sanastoa, on luotava sovitus (mapping) eri ontologioiden välille yhteistoiminnan mahdollistamiseksi tietojärjestelmien välillä. Sovitus erillisten ontologioiden välille luodaan määrittelemällä ontologioiden jakama yhteisenä kielenä (interlingua) toimiva ontologia [Ushold and Gruninger, 2004].

Hybridimalli on kahden edellä kuvatun mallin yhdistelmä. Hybridimallissa, tapaus c) Kuva 5, tietojärjestelmillä on oma paikallinen ontologiansa, mutta paikalliset ontologiat jakavat yhteisen globaalin sanaston, mikä tarjoaa yleisen sanaston sovellusalueella käytettävistä primitiiveistä. Hybridimallin yhteinen sanasto voi olla myös ontologia. Paikalliset ontologiat perustuvat yhteisen sanaston primitiiveihin ja näin ollen yhteistoiminnan saavuttaminen järjestelmien välillä helpottuu.



Kuva 5 Eri tavat käyttää ontologiaa tietojärjestelmissä [Hübner *et al.*, 2001]

Arkkitehtuuri, joka valitaan ontologioiden käyttötavaksi järjestelmässä, on riippuvainen sovellusalueen luonteesta. Yhden ontologian malli on yksinkertainen ja suoraviivaisin toteutukseltaan ja se on myös yksinkertainen hyödyntää, koska kaikki samaa ontologiaa hyödyntävät järjestelmät, seurauksena yhteisen ontologian käytöstä, jakavat saman sanaston ja siis jakavat saman käsityksen kohdealueesta. Ainoastaan yhden ontologian sisältävä malli saattaa olla sopiva lähestymistapa silloin, kun integroitavien järjestelmien käyttämät kuvat sovellusalueesta ovat hyvin lähellä toisiaan. Sitä vastoin voi olla vaikeata määrittellä ontologia riittävän kattavasti ainoastaan yhden ontologian avulla silloin, kun järjestelmien kuva sovellusalueesta on hyvin erilainen. Myös muutokset sovellusalueen käsitteellistämässä aiheuttavat helposti ongelmia single ontologia mallissa, sillä muutosten seurauksena globaali ontologia saattaa muuttua ja sen seurauksena myös ontologian sovittaminen muihin informaatiolähteisiin saattaa vaatia muutoksia. Single ontologiamalliin verrattuna multiple ontologia mallin avulla on helpompi määrittellä paikalliset ontologiat silloin, kun järjestelmien kuva sovellusalueesta on hyvin erilainen. Mutta tällöin saattaa paikallisten ontologioiden välille tarvittavan sovituksen luominen olla työlästä. Hybridimallilla pyritään välttämään single-ontologia ja multiple ontologia mallien ongelmat. Koska hybridimallissa paikalliset ontologiat käyttävät ainoastaan yhteisen jaetun sanaston sanastoa, ovat paikalliset ontologiat edelleen vertailukelpoisina keskenään. [Hübner *et al.*, 2001]

3. Kuvauskielet ontologioiden esittämiseen

XML-kielestä on muodostunut standardi väline tiedon varastointiin ja välitykseen. XML-kielen on tehnyt suosituksi sen suhteellinen yksinkertaisuus ja soveltuvuus tiedon siirtoon järjestelmien välillä sen ollessa laitteistosta ja ohjelmointikielestä riippumaton. XML-kieli on myös suhteellisen joustava soveltuena käytettäväksi sekä ihmisen luettavaksi tarkoitetuissa dokumenteissa että dokumenteissa, jotka on luotu ohjelmistojen käsiteltäväksi. The World Wide Web Consortium (W3C) [W3C, 2006] määrittelee muodollisen kieliopin XML-dokumenteille. XML-dokumentin sanotaan hyvin muotoiltu (well formed), kun se noudattaa XML-kielelle W3C:n määrittelemää yleistä syntaksia ja sääntöjä. XML-kielen voidaan katsoa soveltuvan hyvin informaation välittämiseen, sillä sen rakenne on puhtaasti syntaktinen ja strukturaalinen sekä kuvaa tietoa oliotasolla. XML:n käytön yleisyyden ansiosta ja sen soveltuvuudesta datan välittämiseen on XML-kielestä muodostunut tärkeä väline tietojärjestelmien yhteentoimivuutta tavoiteltaessa. XML kuitenkin määrittelee ainoastaan kielen syntaktisen rakenteen, jolloin täysin identtiset XML-määrittelyt saattavat tarkoittaa hyvin erilaisia asioita siitä riippuen, kuka dokumenttia käyttää. XML:n puutteellisesta kyvystä semanttisen tiedon esittämisestä johtuen on kehitetty kuvauskieliä, joilla voidaan määrittellä dokumenttien semanttinen sisältö.

3.1. Extensible Markup Language (XML)

Yleisen syntaksin ja sääntöjen lisäksi XML-dokumenteille voidaan määrittellä rakenne käyttäen Document Type Definition:a (DTD). DTD:n avulla määritellään XML-dokumentin elementit, jotka ovat sallittuja kyseisessä dokumentissa. Lisäksi DTD:n avulla pystytään määrittelemään dokumentin looginen rakenne määrittelemällä dokumentille sekä sallitut että vaaditut sisäkkäiset elementit. DTD:ta kehittyneempi tapa XML-dokumentin rakenteen kuvaamiseksi on XML scheman käyttö. DTD:hen verrattuna XML schema omaa muutaman tärkeän lisäominaisuuden, joiden ansiosta XML schemaa käyttäen voidaan esittää hyvinkin monimutkaisia tietorakenteita. XML scheman tärkeitä lisäominaisuuksia DTD:hen verrattuna ovat tuki tavanomaisten tietotyyppien käytölle, attribuuttien rajoitukset, hienostuneet rakenteet joita ovat esimerkiksi periytymisen avulla määritellyt tyyppien laajennukset tai rajoitukset, sekä usean scheman yhdistämisen mahdollistava nimiavaruusmekanismi. XML schema itsessään on XML-dokumentti, jolla määritellään XML-dokumentin rakenne. XML-dokumentin sanotaan olevan validi, kun se sisältää joko DTD tai XML schema-määrittelyn ja dokumentti on kyseisen määrittelyn mukainen.

On selvää, että koneet eivät omaa ihmisten ymmärtämystä vastaavaa todellista asioiden ja niiden välisten suhteiden merkityksen ymmärrystä. Haluttaessa koneiden pystyvän käsittelemään eri asioihin liittyviä merkityksiä ihmisten tavoin koodataan asioihin liittyvä ihmisten käsittämä ymmärtämys dokumentteihin siten, että myös koneet pystyvät prosessoimaan koodauksen loogisen päättelyn avulla ihmisten ymmärtämystä vastaavasti. Semanttisen webin dokumenttien semanttisen sisällön kuvaamisen perustana on XML-kieleen perustuvien kuvauskielten käyttö [Berners-Lee *et al.*, 2001]. Kuten aiemmin todettiin, XML-kieli soveltuu ainoastaan tiedon syntaksin ja rakenteen kuvaukseen, joten tietoon liittyvän metatiedon ja merkityksen kuvaamiseen täytyy käyttää jotain muuta tekniikkaa.

3.2. Resource Description Framework (RDF)

Semanttisessa webissä eri resurssien välisten suhteiden kuvaamiseen on otettu käyttöön Resource Description Framework (RDF) [W3C, 2004c]. RDF-kieli on yksi semanttisen webin keskeisimmistä teknologioista, sillä RDF-kielen avulla esitetään semanttiseen webiin sisältyvää metadataa. RDF:n käsitteellinen malli on graafi ja RDF tarjoaa syntaksin kuinka RDF-graafi ilmaistaan käyttäen XML-kieltä. RDF:n ilmaisua XML-kieltä käyttäen kutsutaan RDF/ XML:ksi ja se on normatiivinen muoto ilmaista RDF:ä. RDF-kielessä resurssit, jotka voivat olla esimerkiksi websivuja tai XML-dokumentin elementtejä, esitetään kolmikoiden (triplet) avulla. Kolmikoiden kolme komponenttia ovat resurssi, ominaisuus ja arvo. Nämä kolmikoiden kolme komponenttia voidaan myös tulkita olevan subjekti, predikaatti ja objekti.

Kuvattaessa eri resurssien välisiä suhteita RDF-kielen avulla, jokainen resurssi yhdistetään ominaisuuden arvoon. Ominaisuuden arvo voi olla joko tietotyyppi tai edelleen resurssi. Ominaisuuden arvo voidaan korvata myös muuttujalla, joka edustaa resurssia, ja joka vuorostaan on kuvattu käyttäen sisäkkäisiä kolmikoita. Näin saadaan esitettyä väittämiä kyseisen resurssin ominaisuuksista. RDF-kielessä mahdollistetaan myös moniarvoiset ominaisuudet resursseille. Tätä tarkoitusta varten kielessä on tietorakenne, jonka avulla voidaan määritellä kokoelmia (collections). Mahdolliset kokoelmatyypit ovat järjestämätön lista (bag), järjestetty lista (seq) ja vaihtoehtoiset joukot (alt). Kokoelmat mahdollistavat mekanismin jolla ominaisuuksien arvoja voidaan koostaa. Nimiavaruuksien käyttö RDF-kielessä mahdollistaa, että metadata voidaan koostaa keräämällä viittauksia useasta eri lähteestä ilman, että nimeämisessä tapahtuu törmäyksiä.

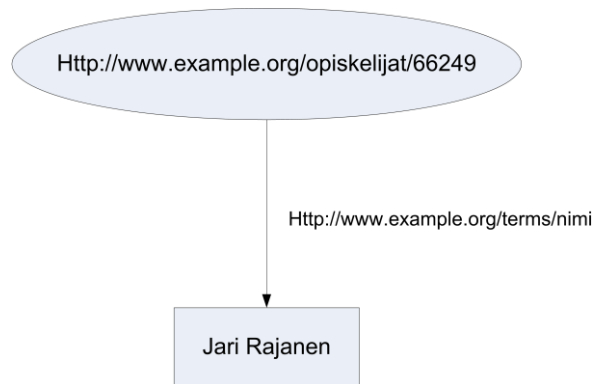
RDF perustuu Uniform Resource Identifiers:n (URI) käyttöön resurssien identifioinnissa. Alla on esimerkki resurssin määrittelystä RDF/ XML-kieltä

käyttäen, jossa määritellään opiskelijan numero '66249' ominaisuuden 'nimi' arvoksi 'Jari Rajanen'.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:extermis="http://www.example.org/termis/">

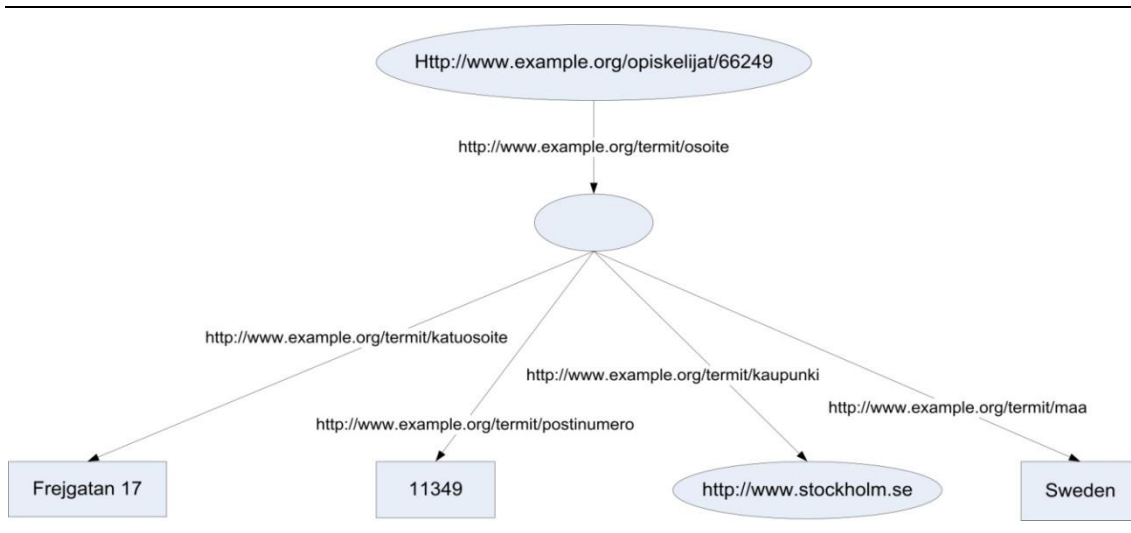
  <rdf:Description
    rdf:about="http://www.example.org/opiskelijat/66249">
    <extermis:nimi>Jari Rajanen</extermis:nimi>
  </rdf:Description>
</rdf:RDF>
```

Kuvattaessa resursseja, niiden ominaisuuksia ja ominaisuuksien arvojen määrittämiä RDF-kieltä käyttäen voidaan RDF-lausekkeet esittää graafien avulla [W3C, 2004c]. Kuva 6 on esimerkki graafista, jossa esitetään yllä oleva esimerkki graafin avulla.



Kuva 6 Esimerkki RDF-graafista

Luonnollisestikaan kaikille mahdollisille asioille ei ole olemassa omaa URI viitettä eikä sitä myöskin ole mielekasta laatia. Tätä varten RDF-kielessä on rakenne, jossa asiat jotka voidaan esittää muun informaation avulla, kuvataan tyhjällä solmulla [W3C, 2004c]. Esimerkki tyhjän solmun käytöstä RDF-graafissa on esitetty Kuva 7, jossa opiskelijan numero 66249 osoite koostuu katuosoitteesta, postinumerosta, kaupungista ja maasta.



Kuva 7 Esimerkki tyhjän solun käytöstä RDF graafissa

Kuva 7 olevassa RDF-graafissa esitetään osoitteen koostuvan muualla esitetystä informaatiosta ja osoite kuvataan tyhjällä solmulla. Alla esitetään graafia vastaava RDF/ XML koodi. Käytettäessä tyhjää solmua RDF/ XML:ssä on solmu identifioitava antamalla solmulle identifioiva viite (nodeId), joka identifioi solmun kyseissä dokumentissa. Toisin kuin URI-viite, tyhjää solmua ei tunneta dokumentin ulkopuolella [W3C, 2004c].

```

<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:exterms="http://www.example.org/termit/">

    <rdf:Description

      rdf:about="http://www.example.org/opiskelijat/66249">
      <exterms:osoite rdf:nodeID="opiskelijanr66249"/>
    </rdf:Description>

    <rdf:Description rdf:nodeID="opiskelijanr66249">
      <exterms:katuosoite>Frejgatan 17</exterms:katuosoite>
      <exterms:postinumero>11349</exterms:postinumero>
      <exterms:kaupunki rdf:resource="http://www.stockholm.se"/>
      <exterms:maa>Sweden</exterms:maa>
    </rdf:Description>

  </rdf:RDF>

```

Samaan tapaan kuin XML scheman avulla voidaan määrittellä rakenne XML-dokumentille, voidaan RDF Scheman avulla määrittellä RDF-dokumentin sanaston merkitys. RDF Scheman määrittelyn erona XML scheman määrittelyyn on, että RDF Scheman avulla ei määrittellä dokumentin syntaksia vaan sen

avulla määritellään dokumentin tietorakenne kuvaamalla luokat, ominaisuudet ja niiden väliset suhteet [Terzi *et al.*, 2003].

RDF-kieltä käyttäen on mahdollista määritellä lausekkeita, joiden avulla pystytään määrittelemään metadataa ja näin RDF-kielen avulla on mahdollista esittää tiedon aiottu merkitys muodossa, jota tietokoneiden on mahdollista käsitellä.

3.3. Web Ontology Language (OWL)

Termien käsitteellisen merkityksen kuvaamiseksi tietojärjestelmissä tarvitaan ontologioita. Ontologioiden kuvaamiseen on kehitetty RDFS-kielestä edelleen kehitetty Web Ontology Language (OWL) [W3C, 2004a]. Rakenteeltaan OWL perustuu XML-kielen käyttöön ja erona esimerkiksi XML schemaan verrattuna on, että OWL kielellä luotu dokumentti on tietämyksen esittämistä eikä viesti [W3C, 2004b]. OWL-kielellä kuvataan formaaleja sanastoja kuvauslogiikan (description logic) avulla. Kuvauslogiikat ovat ensimmäisen kertaluvun predikaattilogiikan osajoukkoja, joille on onnistuttu kehittämään ratkeavia ja tehokkaita algoritmeja. Ensimmäisen kertaluvun logiikan käytön ansioista OWL-kieli soveltuu hyvin tietämyksen, jota tietokoneiden on tarkoitus ymmärtää, esittämiseen. Loogisten konnektiivien avulla luodaan päättelysääntöjä, joiden avulla tietokone pystyy päättämään asioiden tarkoitetun merkityksen. Ontologioiden automaattista käsittelyä varten on kehitetty päättelykoneita (reasoner).

OWL-kielessä käsitteet esitetään pääsääntöisesti luokkina (class) ja käsitteiden esiintymät yksilöinä (individuals), OWL-kieli perustuu avoimen maailman olettamukseen (open world assumption). Sen perusteella ontologia voidaan määritellä yhdestä tai useammasta lähteestä. Seurauksena rakenteesta on, että sama luokka voidaan määritellä useassa eri lähteessä. Eri lähteistä saatu luokkaa koskeva informaatio voi olla toisilleen vastakkaisia väitteitä; kuitenkin luokkaa koskevaa informaatiota voidaan ainoastaan lisätä keräämällä sitä eri lähteistä, sitä ei voida koskaan poistaa. Ontologian koostamiseen useasta eri lähteestä liittyy myös tärkeä ominaisuus vastaavuuksien määrittelyyn. Eri nimillä määritellyt yksilöt voidaan linkittää määreellä `owl:sameAs` tarkoittavan samaa yksilöä. [W3C, 2004b]

OWL-kielestä on kolme eri standardoitua muotoa: OWL Lite, OWL DL ja OWL Full, jotka eroavat toisistaan kuvausvoimansa suhteen. OWL-kielen eri versioista OWL Lite ja OWL DL ovat laskennallisesti ratkeavia. Kuten nimestä voidaan päätellä, on OWL Lite kuvausvoimansa puolesta kevyin ja OWL Full kuvausvoimansa puolesta monipuolisin. OWL-kielen eri versioilla luodut eri ontologiat ovat ylöspäin yhteensopivia siten, että laillinen OWL Lite -kielellä

luotu ontologia on myös laillinen OWL DL ontologia. Ja laillinen OWL DL ontologia on laillinen OWL Full ontologia. Alaspäin yhteensopivuus ei päde. [W3C, 2004b]

OWL-kieltä käytettäessä valittavan muodon valinta perustuu tarvittavan ilmaisuvoiman määrään. OWL Lite tarjoaa kevyimpänä parhaan laskennallisen tehokkuuden, mutta esimerkiksi kardinaliteettien sallitut arvot ovat ainoastaan 0 ja 1 [W3C, 2004b]. OWL DL laajentaa OWL Litea muun muassa loogisten konnektiivien suhteen ja mahdollistaa uusien luokkien luomisen niiden avulla. OWL Full on ilmaisullisesti joustavin ja mahdollistaa esimerkiksi luokan käsittelyn samanaikaisesti sekä joukkona yksilöitä että omana yksilönään. OWL Full on täysin yhteensopiva RDFS-kielen kanssa mutta ei ole laskennallisesti ratkeava.

4. Yrityksen tietojärjestelmien yhteistoiminnallisuus

Informaation integroimiseksi on yrityksissä tehty paljon töitä. Järjestelmiä on kehitetty erilaisiin tarpeisiin ja eri konteksteihin. Tästä johtuen eri järjestelmillä ja niitä kehittäville ja käyttäville ihmisillä on erilaisia näkökulmia sovellusalueilta ja myös siitä, mikä on oleellista asioita käsiteltäessä ja esitettäessä. Järjestelmissä saatetaan käyttää eri termejä samoista asioista. Myös järjestelmien käyttämissä käsitteistöissä, rakenteissa ja metodeissa saattaa olla eroavaisuuksia niiden ollessa joko päällekkäisiä tai ne eivät vastaa toisiaan [Uschold and Gruninger, 1996]. Tästä johtuen eri järjestelmillä on vaikeuksia ymmärtää toisiaan ja niiden integrointi on vaikeaa.

Yhteistoiminnallisuus on avain menestyksekkääseen tiedon jakamiseen tietojärjestelmien välillä. Stuckenschmidt *et al.* [2000] mukaan onnistuneesta informaation integroinnista saatavat hyödyt ovat ilmeisiä:

- Tiedon laatu paranee laajan ja täydellisen tiedon saatavilla olon ansiosta.
- Kustannusten lasku sen ansiosta, että useat eri tahot hyödyntävät olemassa olevia tiedon lähteitä.
- Tarpeettoman tiedon ja konfliktien välttäminen, jotka saattavat aiheuttaa tarpeettomasti kahdennetusta tiedosta.

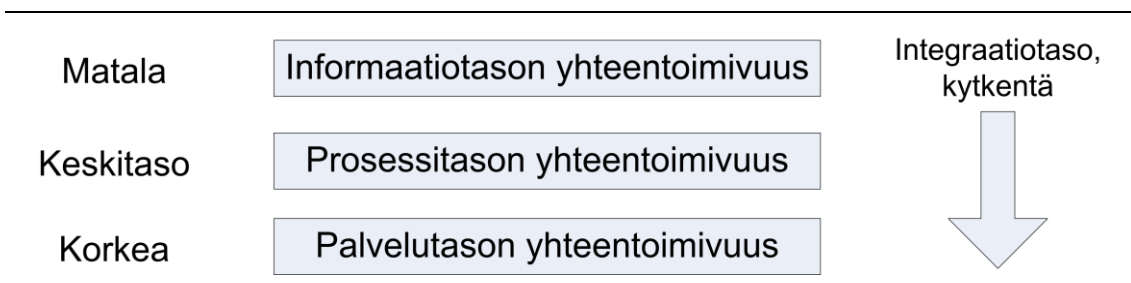
Järjestelmien jakaessa samat käsitykset sovellusalueesta muodostaa se pohjan keskinäisten suhteiden, prosessien ja sovellusalueen muodolliselle määrittelylle. Formaalin määrittelyn käyttö luo mahdollisuuksia yhteisten ohjelmistokomponenttien käytölle ja ohjelmistokomponenttien uusiokäytölle [Uschold and Gruninger, 1996]. Muodollinen määrittely mahdollistaa myös informaation yhtenäisyyden automaattisen tarkistamisen, mikä tukee sovellusten luotettavampaa toimintaa.

4.1. Tietojärjestelmien yhteistoiminnallisuuden tasot

Tavoiteltaessa yhteistoiminnallisuutta itsenäisten tietojärjestelmien välillä on välttämätöntä, että järjestelmä jolle tieto on annettu saataville, kykenee myös prosessoimaan ja tulkitsemaan tietoa virheettömästi. Pyrittäessä yhteistoiminnallisuuteen heterogeenisten tietojärjestelmien välillä on ratkaistava ongelmia usealla eri tasolla. Tarkasteltaessa järjestelmien yhteistoiminnallisuutta voidaan keskittyä puhtaasti tarkastelemaan millä tasolla tietojärjestelmät ymmärtävät toisten tietojärjestelmien dataa, tietoa ja tietämystä. Jotta järjestelmien yhteistoiminnallisuus on mahdollista, on ratkaistava yhteensopivuusongelmat syn-

taktisella, rakenteellisella ja semanttisella tasolla [Stuckenschmidt *et al.*, 2000]. Johannesson *et al.* [2005] esittävät edellä esitettyyn luokitteluun nähden hieman eri näkökulmasta tehtävän tavan luokitella tietojärjestelmien välisen yhteistoiminnallisuuden tasoa. Heidän esittämä luokittelu on sovelias tarkasteltaessa erityisesti itsenäisten tietojärjestelmien yhteistoiminnallisuutta eri yritysten välillä.

Johannesson *et al.*:n [2005] esittämää yhteistoiminnallisuuden tason luokittelua on havainnollistettu Kuva 8. Heidän esittämänsä luokittelun alimmalla yhteistoiminnallisuuden tasolla itsenäisesti toimivat yrityksen (yritysten) tietojärjestelmät on integroitu yksinkertaisella rakenteella, jonka avulla on mahdollista tuoda (import) ja viedä (export) dataa tai informaatiota järjestelmään (järjestelmästä). Integraation keskitasolla yritykset koordinoivat yhteisiä liiketoimintaprosesseja tarkoituksena saavuttaa yhteinen ymmärtämys keskinäisestä vuorovaikutuksesta. Korkeimmalla yhteistoiminnallisuuden tasolla toimii yritysten tietojärjestelmien yhteistoiminnallisuus palvelutasolla, jossa yritykset määrittelevät julkiset rajapintansa eri palveluille käyttäen teollisuusstandardeja, kuten esimerkiksi EDI ja ebXML elektronisen kaupankäyntinsä toteuttamiseksi.



Kuva 8 Tietojärjestelmien yhteistoiminnallisuuden tasot Johannesson *et al.* [2005] mukaan

Palvelutason yhteistoiminnallisuudessa tarkastellaan yrityksen tietojärjestelmää liiketoimintaprosessin näkökulmasta. Tällöin liiketoimintaprosessin katsotaan muodostuvan eri palveluista, joita hyödynnetään tuottamaan liiketoimintaprosessin muodostavien aktiviteettien tarvitsema toiminnallisuus.

4.1.1. Syntaktinen ja rakenteellinen yhteistoiminnallisuus

Tyypillinen syntaktisella tasolla oleva eri järjestelmien välinen yhteensopivuusongelma on esimerkiksi toisistaan poikkeavien tietotyyppien käyttö integroitavissa järjestelmissä. Yleinen ratkaisu ongelmaan on kapselointi, jolloin tietolähteen sisäinen rakenne kätketään kääreen sisään ja tiedon jakamista varten ulospäin tarjotaan integroitaville järjestelmille yhteisen tietomallin mukainen rajapinta.

Heterogeenisten tietojärjestelmien yhteensovittamisessa on myös usein ongelmana tiedon rakenteen erilaisuus integroitavissa järjestelmissä, jolloin on tarpeen muotoilla tietorakenteet homogeeniseen muotoon. Ratkaisuna on usein väliohjelmiston (middleware) käyttö [Stuckenschmidt *et al.*, 2000].

4.1.2. Semanttinen integrointi

Tietokantajärjestelmiä tarkasteltaessa tarkastelijalla on usein selvillä tapa kuinka tieto on tietokannassa organisoitu ja mikä on tiedon tarkoitettu merkitys, käsitteellistäminen. Tietokannan rakenne kuvataan tietokantakaaviossa ja käsitteellistäminen on sisällytetty metadataan. Tietokantajärjestelmien yhteistoiminnallisuus semanttisella tasolla edellyttää, että integroitavat järjestelmät tulkitsevat tietoa samaa merkitystä käyttäen; järjestelmät käyttävät tietoa yhtenevässä semanttisessa merkityksessä. Kahden itsenäisen tietokantajärjestelmän jakaessa tietoa keskenään emme voi olla varmoja tiedon oikeasta tulkinnasta ennen kuin tiedämme, että järjestelmien käyttämän tiedon käsitteellistäminen on yhteneväinen.

Esimerkiksi kahdessa eri tietokannassa voidaan jossain tietokannan kentässä esittää numeerista dataa jolla ilmaistaan pituusmittaa. Kuitenkin toisessa tietokannassa voi pituus olla ilmaistuna senttimetreinä ja toisessa tuumina. Eli tietokantojen kentissä ei ole pituuden mittaa ilmaistu käyttäen samaa semanttista merkitystä.

4.2. Ontologiat ja tietojärjestelmien yhteistoiminnallisuus

Useassa eri lähteessä esitetään, että ontologioita voidaan hyödyntää yrityksen tietojärjestelmien yhteistoiminnallisuuteen. Vaikka yleisesti on hyväksytty näkökanta, että ontologioiden käyttö voi tukea tietojärjestelmien yhteistoiminnallisuutta, vaikuttaa kuitenkin olevan epävarmaa kuinka ja miten ontologioita voidaan tarkalleen ottaen käyttää yhteistoiminnallisuuden saavuttamiseksi. Visiossaan semanttisesta elektronisesta liiketoiminnasta Sing *et al.* [2005, p.38] sanovat:

Vaaditaan paljon työtä ymmärtämään kuinka käsitteellistämiset (conceptualizations), jotka muodostavat liiketoimintaprosesseja yrityksen piirissä, voidaan vangita, esittää, jakaa ja prosessoida sekä ihmisten että älykkäiden agenttien toimesta. Tämä pyrkimys tulee loppujenlopuksi johtamaan läpinäkyvään ja turvalliseen informaatioon ja tietämyksen virtaan palvelu- ja toimitusketjuissa, ja tulee lisäämään taloudellista tehokkuutta digitaalisessa taloudessa.

Tarkasteltaessa ontologioiden hyödyntämistä tietojärjestelmien yhteistoiminnallisuuteen, niin yhtenä tärkeänä periaatteena on, että jos kaksi tai use-

ampi järjestelmä jakaa saman ontologian, niin pystytään päättämään tapaukset milloin järjestelmät käyttävät tietoa samassa semanttisessa merkityksessä; jos järjestelmät käyttävät ontologioita, jotka eivät ole vertailukelpoisia keskenään, ei kyetä päättämään niiden käyttävän tietoa samassa merkityksessä tilanteessa, jossa eri järjestelmien tiedot vastaavat semanttisesti toisiaan tiedon ollessa vain vertailukelvottomien ontologioiden avulla määritelty. Visiossa semanttisesta webistä mikä tahansa tietämystä hyödyntävä järjestelmä on kykenevä integroitumaan webiin, jolloin järjestelmät pääsevät käsiksi informaatioon ja ymmärtävät tiedon semanttisen merkityksen [Berners-Lee *et al.*, 2001]. Nykyisin on laajasti hyväksytty ajatus webissä toimivista ohjelmista, jotka kykenevät yhteistoiminnallisuuteen toisten itsenäisten ohjelmien kanssa ja pystyvät hyödyntämään toistensa dataa. Ontologioiden käyttö on yksi semanttisen webin avainteknologioista, jonka pääasiallisena tarkoituksena on mahdollistaa tietojärjestelmien yhteistoiminnallisuus järjestelmien rakentamisessa käytetystä teknologiasta, arkkitehtuurista tai sovellusalueesta riippumatta. Semanttisen webin toimintaperiaate on suhteellisen yksinkertainen. Osapuolet joilla on tarjolla ohjelmisto, dataa tai palvelu identifioivat jonkin yleisen käsitteellistämisen datalle, mikä täsmennetään mahdollisimman hyvin; järjestelmien yhteistoiminnallisuus perustuu edellä mainittuun käsitteellistämisen määrittelyyn [Gruber, 2007]. Berners-Lee *et al.*:n esittämän vision semanttisesta webistä pohjalta voidaan ajatella, että semanttisen webin teknologioita voidaan hyödyntää tietojärjestelmien yhteistoiminnallisuuteen muissakin yhteyksissä, kuten esimerkiksi yritysten tietojärjestelmien yhteistoiminnallisuudessa.

Johannesson *et al.* [2005] suosittelevat omassa työssään ontologioiden hyödyntämistä kahdella eri taholla: tietämyskantana edistämään ihmiseltä ihmiselle tapahtuvaa tietämyksen jakamista sekä mahdollistamaan ihmisen käsitteellistämisen tietämyksen siirtämistä tietokoneelle siten, että tietokone pystyy käsittelemään tietämystä ihmisen käsityskyvyn kaltaisesti.

5. Ontologiat ja heterogeenisten tietokantojen yhteistoiminnallisuus

Yritysten tietojärjestelmissä käytettävä tieto on useimmiten tallennettu tietokantoihin. Guarino [1998] toteaa informaation integroinnin olevan merkittävä käyttökohde ontologioille. Hän toteaa edelleen, että ilmeinen ontologioiden käyttökohde ovat tietokannat, joissa ontologiaa voidaan verrata tietokantakaavioon. Voitaneen ajatella, että mikäli tietokantaan tallennettu tieto kuvataan käyttäen ontologiaa, joka on vertailukelpoinen tietokantaa hyödyntävässä tietojärjestelmässä käytettävän ontologian kanssa, on tietokannan hyödyntäminen selkeää, jos ontologian avulla voidaan varmistua tietokantaa käyttävän järjestelmän käyttävän tietoa yhtenevässä semanttisessa merkityksessä tietokannan kanssa. Luotaessa uusia tietojärjestelmiä hyödynnetään usein olemassa olevia tietokantoja. Uutta tietojärjestelmää rakennettaessa saattaa olla suureksi hyödyksi, jos tietojärjestelmän käyttäessä useita eri tietokantoja tietokannat pystyvät yhteistoimintaan keskenään [Rahm and Bernstein, 2001]. Yhteistoiminnallisia tietokantoja hyödynnettäessä voidaan saavuttaa merkittäviä säästöjä tietojärjestelmän kehitystyössä, mikäli tietokantojen käyttöönotto ja tiedon semanttisen sisällön varmentaminen voidaan suorittaa (puoli)automaattisesti.

Guarinin [1998] mukaan ontologioita voidaan hyödyntää sekä tietojärjestelmän kehitysvaiheessa että käytön aikana. Kehitysvaiheessa ontologia voi merkittävästi tukea vaatimusmäärittelyä sekä käsitteellistä mallintamista. Järjestelmän ollessa käytössä ontologia voi olla ratkaisu tietokantajärjestelmien yhteistoimintaan.

Yleensä jokaisella tietokannalla on oma kaavio, joka kuvaa tietokannan rakenteen. Jos kaksi tietokantaa jakaa ontologian joka määrittelee tietokannan kaavion niin silloin tietokannat jakavat myös saman käsitteistön, koska ontologia on käsitteellistämisen erikoistamista, kuten aiemmin on mainittu. Guarino [1998] sanoo, että vaikka kaksi tietokantajärjestelmää jakaa saman sanaston, emme voi olla varmoja että ne käyttävät tietoa samassa merkityksessä jolleme tiedä, että järjestelmät käyttävät samaa käsitteistöä. Ontologioita voidaan hyödyntää tietokantajärjestelmien yhteentoimivuuden saavuttamisessa käyttämällä ontologioita tietokantakaavion määrittelyyn.

Guarinin [1998] mukaan ontologiaa voidaan verrata tietokannan kaavio komponenttiin. Guarinin mukaan ontologiat voivat olla hyödyksi tietokantojen yhteistoiminnan saavuttamisessa sekä tietokantojen suunnitteluvaiheessa että käytön aikana.

5.1. XML kaaviot

XML-dokumenttien käyttö tiedon välityksessä on laajalle levinnyt sovellus tiedon jakamiseksi itsenäisten tietokantajärjestelmien välillä. Tiedon siirrossa tietokantojen välillä on kaksi mahdollista toimintamallia. Yksi toimintamalli on, että otettaessa vastaan XML-dokumentti joka sisältää tietoa mikä halutaan tallentaa tietokantaa, käytetään jäsenintä (parser) poimimaan haluttu tieto XML-dokumentista ja haluttu tieto tallennetaan normaaliin tapaan tietokantaan. XML-dokumenttia ei säilytetä. Kun vastavuoroisesti tarvitaan tietoa tietokannasta, haetaan tarvittava tieto tietokannasta ja luodaan uusi XML-dokumentti. Toinen mahdollinen toimintamalli on käyttää XML tietokantaa. XML tietokanta ymmärtää suoraan XML-dokumentin rakenteen ja pystyy käsittelemään suoraan XML-dokumenttia. XML kaaviota käyttäen voidaan määritellä validi XML-dokumentti ja hyödyntää sitä järjestelmien integroinnissa. Yksi mahdollinen sovellus on, että kaikki yhteistoimintaan osallistuvat tietokannat käyttävät samaa standardia XML kaaviota. Standardia XML kaaviota käyttäen voidaan saavuttaa sekä syntaktinen että rakenteellinen yhteistoiminta. XML kaaviota käyttäen ei voida kuitenkaan määritellä tiedon aiottua merkitystä. Joten edelleen saattaa ongelmana olla tietokantojen yhteistoiminnan ongelma semanttisella tasolla [Stuckenschmidt *et al.*, 2000]. On lähes väistämätöntä, että tietokantoja muokataan ajan mittaan, joten myös XML kaaviota joudutaan muokkaamaan. Standardia XML kaaviota käytettäessä saattaa XML kaavion muokkaus tuoda esiin ongelmia XML kaavion versioiden hallintaan liittyen. Lisäksi saattaa vaikeutena olla varmistua siitä, että kaikki järjestelmät tulkitsevat tietoa samalla tavoin, käyttävät tietoa samassa semanttisessa merkityksessä.

Monessa tietokantoja käyttävissä hajautetuissa tietojärjestelmissä ohjelmistojen välinen tiedonsiirto toteutetaan nykyisin standardoitua XML kaaviota käyttäen. Standardoidun XML kaavion käyttö mahdollistaa ohjelmistojen välisen tiedonsiirron yhteensovittamisen. Tietokantoja joudutaan väistämättä muuttamaan ajan myötä ja käytettäessä standardoitua XML kaaviota tiedonvälityksessä ovat ongelmana kaavion eri versioiden hallinta sekä varmistautuminen siitä, että yhteistoiminnassa olevat järjestelmät käsittelevät tietokannan elementtejä samaa käsitteellistä merkitystä käyttäen. Dou ja LePendou [2006] osoittavat, että tietokantakaavio voidaan esittää yksinkertaisia muunnoksia käyttäen ontologiana. Korvaamalla standardoitu XML kaavio käytön aikana saatavilla olevalla ontologialla voidaan varmistaa yhteistoiminnassa olevien tietojärjestelmien käyttävän elementtejä samassa käsitteellisessä merkityksessä sekä voidaan mahdollisesti automatisoida tietokannan rakenteen evoluution hallinta [Dou and LePendou, 2006].

5.2. Ontologioiden käyttö tietokantojen suunnittelussa

Tietokantajärjestelmää kehitettäessä voivat ontologiat olla hyödyksi vaatimusmäärittelyssä ja käsitteellisessä mallintamisessa [Guarino, 1998], kun tavoitteena on itsenäisten tietokantojen yhteistoiminnallisuus. Jos tietokantaa suunniteltaessa on saatavilla ylimmän tason ontologia, niin ontologia voi tarjota yleisen tason käsitteellisen mallin, mikä tukee heterogeenisten tietokantojen integrointia. Kun tietokantojen, joiden on tarkoitus jakaa tietoa keskenään, suunnittelussa käytetään yhteistä yleisen tason käsitteellistä mallia, niin on mahdollista varmistaa, että tietokannoissa käytetään samaa käsitteellistä merkitystä tallennettavasta tiedosta.

5.3. Ontologioiden hyödyntäminen käytön aikana

Jos tietokantakaavio on määritelty käyttäen käytön aikana saatavilla olevaa ontologiaa, se voi tarjota tietokannan kaavion joka toimii välittäjänä itsenäisten tietokantojen yhteentoimivuuden saavuttamiseksi. Yksi mahdollinen ratkaisu heterogeenisten tietokantojen yhteistoiminnallisuuteen on käyttää globaalia tietokantakaaviota, mikä on määritelty käyttäen formaalia ontologiaa [Aparicio *et al.*, 2005]. Käytön aikana voi tietokantajärjestelmän sovelluskerros ratkaista entiteettien väliset suhteet perustuen siihen faktaan, että kaikki tietokannat jakavat saman terminologian, koska se on peritty yleisestä ontologiasta.

5.4. Ontologioiden käyttö tietämyksenhallinnassa

Ontologiat ovat pohjimmiltaan olemassa olevan tietämyksen esittämistä, joten voidaan ajatella ontologioiden käytön sopivan tietämyksenhallinnan sovelluksiin. Mallinnettaessa jotakin kohdealuetta ontologian avulla esitetään käsitteet luokkina ja niiden välisinä suhteina. Esittämällä ontologiassa olevista käsitteistä reaali maailman ilmentymiä, voidaan todellisia faktoja ja tiedon kappaleita tallentaa ontologiaan [Brandt *et al.*, 2008]. Brandt *et al.*:n [2008] mukaan ontologioiden käytöllä on kaksi pääasiallista etua perinteisiin tietokantakaavioihin verrattuna:

- Ontologiat ovat äärimmäisen joustavia. Tietorakenteita voidaan muokata ja laajentaa jopa käytön aikana.
- Ontologiat voidaan esittää koneiden luettavissa olevalla logiikkaan perustuvalla kielellä, mikä mahdollistaa käsitteisiin liittyvän semantiikan määrittelyn. Tästä on seurauksena, että tietokoneet voivat tulkitä ja tehdä päätelmiä ontologiaan tallennetun tiedon perusteella. Tämän johtaa kehittyneempään tiedon hallintaan ja tiedonhakuun. Ontologiasta on esimerkiksi mahdollista suorittaa semanttinen tie-

donhaku, mikä verrattuna yksinkertaisempaan, esimerkiksi avainsanahakuun, mahdollistaa systemaattisemman tiedon noutamisen.

Ontologia voidaan käyttää tietämuskantana, sillä se sisältää termit joita tarvitaan sovellusalueesta puhuttaessa. Lisäksi luotaessa ontologiaa voidaan ontologiassa kuvata käsitteiden ilmentymiä ja määrittellä säännöt niiden välisille suhteille. Esitettäessä sovellusalueen tietämys ontologian avulla mahdollistetaan kohdealueen tietämyksen analysointi tietokoneita käyttäen.

6. Tietojärjestelmäintegraatiosta tietojärjestelmien yhteistoiminnallisuuteen

Tietojärjestelmäintegraatiolla (Enterprise Application Integration ((EAI)), tarkoitetaan yritysten eri tietojärjestelmien yhteensovittamista. Useat yritysten liiketoimintaprosesseista koskettavat useampaa yrityksessä käytettyä sovellusta. Näiden sovellusten tulisi pystyä kommunikoimaan keskenään reaaliajassa, jotta yritys pystyisi toimimaan tehokkaasti ja kommunikoimaan ulkomaailman kanssa. Tyypillisesti yrityksissä on vuosien saatossa syntynyt sovellussaarekkeitä eri järjestelmistä, kuten esimerkiksi toiminnanohjaus (enterprise resource planning (ERP)), asiakkuuden hallinta (customer relationship management (CRM)) ja toimitusketjun hallinta (supply chain management (SCM)), mitkä eivät pysty suoraan keskustelemaan keskenään. Uusien monipuolisten korkean tason sovellusten luominen edellyttää usein, että eri järjestelmiä joudutaan integroimaan keskenään. Tietojärjestelmäintegraation tavoitteena on sovittaa yhteen ja rationalisoida liiketoimintaprosesseja itsenäisten sovellusten ja liiketoimintayksiköiden välillä [Bouras *et al.*, 2006].

Sovellusten integrointia voidaan pitää kuitenkin vain osittain tyydyttävänä ratkaisuna luotaessa monimutkaisia sovelluksia eri järjestelmiä yhdistämällä. Kukin integroitavista järjestelmistä omaa käytännössä oman näkemyksensä todellisuudesta, ja jotta järjestelmien välillä voidaan jakaa tietoa edellyttää se käytännössä aina käsityönä tehtävää monimutkaisten rakenteiden luomista järjestelmien välille, jotta pystytään ratkaisemaan kohdealueen esitystapaan, semanttisiin yksityiskohtiin, tietämyksen rakenteeseen ja ohjelmallisiin vakioihin liittyvät ongelmat [Pollock, 2001]. Yhteistoiminnalliset tietojärjestelmät pystyvät rutiininomaisesti jakamaan tietoa keskenään ja hyödyntämään toistensa tietoa.



Kuva 9 Sovelluksen yhteistoiminnallisuus [Pollock, 2001]

Kuten Kuva 9 nähdään, sovelluksen yhteistoiminnallisuus muodostuu sekä informaation että sovelluksen integroinnista. Perinteisissä EAI-ratkaisuissa pyritään luomaan yhtenäinen toimintatapa ja kuva yrityksestä integroimalla liiketoimintaprosesseja ja sovelluksia viestien ja tapahtumien tasolla, kun taas yhteistoiminnallisuudessa pyritään mahdollisimman pieneen informaation vaihtamisen tarpeeseen järjestelmien yhteistoiminnallisuuteen pyrittäessä [Pollock, 2001].

6.1. Väliohjelmistoista Web service –tekniikan käyttöön tietojärjestelmäintegraatiossa

Perinteinen tietojärjestelmäintegraatio perustuu väliohjelmistojen (middleware) käyttöön integroitavien järjestelmien välillä, missä viestien lähettäminen järjestelmien välillä, sovellusrajapintoja käyttäen, toteutetaan liittämällä yhteen eri väliohjelmistoratkaisuja ja kehittämällä erilaisia adaptereita, liittimiä ja lisäosia. Uuden sovelluksen lisääminen järjestelmään tarkoittaa yleensä aina sovelluksen räätälöintiä, joka on sekä aikaa vievää että kallista. Lisäksi luotaessa uusia korkeamman tason sovelluksia integroimalla eri järjestelmiä törmätään uuteen ongelmaan; perinteiset tietojärjestelmäintegraatoratkaisut ovat toimittajakohtaisia, joten tällöin joudutaan integroimaan eri EAI-sovelluksia. Suurimpana ongelmana perinteisessä tietojärjestelmäintegraatiossa on korkeamman abstraktiotason sekä standardoitujen arkkitehtuurien ja toteutusten puuttuminen [Bouras *et al.*, 2006]. Palvelusuuntautuneen arkkitehtuurin (service-oriented architecture (SOA)) esittely ja siihen perustuvan Web service -tekniikan kehittyminen on mahdollistanut uuden lähestymistavan syntymisen tietojärjestelmäintegraatioon. Web service -tekniikan hyödyntäminen ja SOA-arkkitehtuuri ovat teollisuuden trendejä tällä hetkellä [RIDE Consortium,

2008]. Web service -teknologian avulla on voitu ratkaista edellä kuvattuja perinteisen tietojärjestelmäintegraation ongelmia, sillä perinteiseen tietojärjestelmäintegraatioon verrattuna Web service -teknologia tarjoaa korkeamman abstraktiotason ohjelmistojen rakenteeseen. Se muodostaa uuden abstraktin kerroksen olemassa olevien ohjelmistojen yläpuolelle ja kytkee XML-kielen käytön ohjelmien, objektien, tietokantojen ja kokonaisvaltaisten ohjelmistojen väliseen kommunikointiin. Tämä korkea abstraktiotaso piilottaa täysin pohjalla käytetyt ohjelmointitekniikat. Juuri tähän abstraktiin kerrokseen sekä laajalle levinneiden ja yleisesti hyväksytyjen standardisoitujen protokollien käyttöön perustuu mahdollisuus hyödyntää Web service -teknologiaa ohjelmakomponenttien ja ohjelmistojen integrointiin.

6.2. Web services –teknologian käyttö tietojärjestelmäintegraatioon

Web services on W3C:n standardoima palvelusuuntautunut arkkitehtuuri. Luotaessa tietojärjestelmiä yritysten käyttöön palvelusuuntautunutta arkkitehtuuria käyttäen, voidaan mikä tahansa toiminnallisuus esittää järjestelmässä saatavilla olevana palveluna. Palvelusuuntautuneessa arkkitehtuurissa tuetaan löyhää sidontaa eri ohjelmistokomponenttien välillä, mikä mahdollistaa tietojärjestelmissä käytettyjen komponenttien uusiokäytön luotaessa uusia järjestelmiä ja siten on olemassa mahdollisuus tulevaisuudessa kustannusten pienentymiseen komponenttien uusiokäytön myötä [Bouras *et al.*, 2006]. Web services –ohjelmiston määrittämällä olevan yhteistoiminnallisuuden mahdollistava ohjelmisto, joka on suunniteltu tukemaan verkossa tietokoneelta tietokoneelle tapahtuvaa vuorovaikutusta [W3C, 2004d]. Web services -teknologia sopii käytettäväksi sovellusten integrointiin sekä organisaation sisäisesti että organisaatioiden välillä.

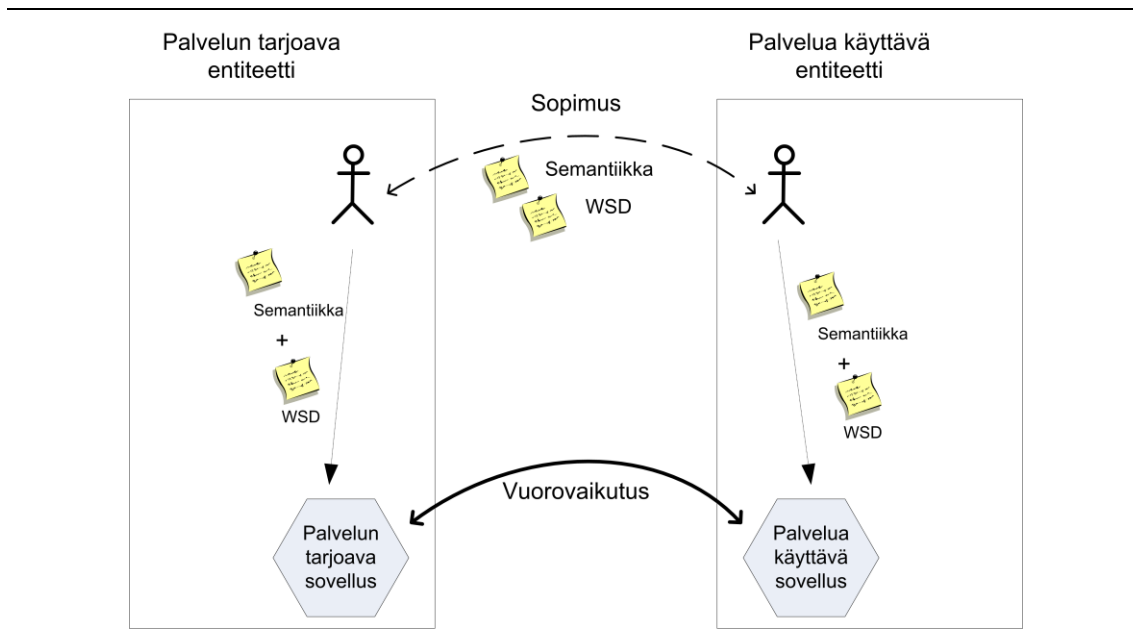
6.2.1. Web services arkkitehtuuri

Web services –teknologiaa käyttäen toteutetut sovellukset ovat itsenäisiä modulaarisia sovelluksia, joita voidaan kutsua verkon yli. Web services -teknologialla toteutettavat palvelut voivat vaihdella yksinkertaisen kyselyn vastauksesta monimutkaiseen liiketoimintaprosessiin. Palvelurajapinta Web services -palveluihin on kuvattu tietokoneiden tulkittavissa olevassa muodossa käyttäen Web Service Description Language:a (WSDL) tarjoten alustasta riippumattoman kuvauksen ja pääsyn informaatioon taustajärjestelmän toiminnallisuudesta. Vuorovaikutukseen Web service -sovelluksen kanssa käytetään palvelunkuvauksen mukaisia SOAP-viestejä yhdessä muiden standardoitujen web-teknologioiden kanssa.

Luotaessa uusia toimintoja yrityksen tietojärjestelmään laatimalla tarvittaessa uusia palveluita tarjoavia sovelluksia tai yhdistelemällä olemassa olevia palveluita tarpeen mukaan, ei Web service -teknologiaa käytettäessä tarvita mitään erillisiä sovittimia tai väliohjelmistoja, koska vuorovaikutus palvelun tarjoajan ja käyttäjän välillä on standardoitu käyttämällä SOAP-viestejä. Palvelun käyttäjän tulee ainoastaan selvittää tarvittavan palvelun sisältö, palvelurajapinnan käyttötapa sekä tarvittavan palvelun sijainti. Käytettäessä organisaation sisäisiä palveluita on palvelun käyttäjällä usein selvillä tarvittavan palvelun sijainti sekä palvelurajapinnan kuvaus. Web Services -arkkitehtuurin mukaisesti voi palvelun tarjoaja myös rekisteröidä tarjoamansa palvelun Universal Description, Discovery, and Integration (UDDI) -rekisterissä kuvaamalla tarjoamansa palvelun kuvauksen käyttäen WSDL-kieltä, jotta palvelua tarvitsevien on mahdollista tutkia palvelun kuvausta ja paikallistaa tarjolla oleva palvelu [Bouras *et al.*, 2006]. Vaikka UDDI-rekisterissä oleva palvelun kuvaus on tietokoneiden luettavassa muodossa WSDL-kielen käytön ansiosta, ovat nykyiset WSDL-kuvaukset kuitenkin puhtaasti syntaktisia [Bruijn *et al.*, 2005]. UDDI-rekisteri tallentaa kuvauksia eri palveluista kategorisoiden niitä taksonomeihin ja tarjoaa palvelun tarvitsijalle mahdollisuuden etsiä sopivia palvelukuvauksia hakemistopalvelusta [Bouras *et al.*, 2006].

6.2.2. Web services -teknologian käyttö

Halutessaan käyttää jotakin Web service -palvelua palvelun käyttäjän täytyy ensin löytää sopivan palvelun tarjoaja. Palvelun paikallistaminen nykyisellään tapahtuu paljolti manuaalisesti, tyypillisimmin järjestelmän suunnitteluvaiheessa. Palvelun käyttäjä saa palvelun paikallistamisen myötä käyttöönsä WSDL:lla määritellyn palvelun kuvauksen. Palvelun kuvaukseen liittyy myös kuvaus palvelun toiminnallisuudesta (functional description). Web services -arkkitehtuurin mukaisen järjestelmän prosesseja on havainnollistettu Kuva 10. W3C:n määritelmän mukaisesti tulee sekä palvelun tarjoajan ja käyttäjän sitoutua omalta osaltaan sekä tuottamaan että käyttämään omassa roolissaan palvelua palvelukuvauksen mukaisesti. Oleellista manuaalisessa tapahtuvassa palvelun valinnassa on, että palvelukuvauksen semanttisen sisällön tulkinta tehdään ihmisvoimin. Kun on varmistauduttu, että molempien osapuolten vuorovaikutukseen osallistuvat sovellukset käyttävät yhtenevää semanttista tulkintaa sekä palvelun kuvauksesta että sen toiminnallisuudesta, voivat palvelua kutsuva ja palvelun tarjoava sovellus vaihtaa SOAP-viestejä keskenään. [W3C, 2004d]



Kuva 10 Prosessit Web services –arkkitehtuurissa [W3C, 2004d]

Web service -arkkitehtuurin mukaisten palveluiden paikallistaminen ja palveluiden hyödyntämiseksi tarvittavan sidonnan muodostaminen on mahdollista tehdä dynaamisesti esimerkiksi UDDI rekisteriä käyttäen. Kuitenkaan ei uusien palveluiden automatisoitu käyttöönotto ole mahdollista perinteisessä Web service ympäristössä. Perinteiset Web service -palveluiden kuvaukset on tarkoitettu ihmisten käsiteltäväksi ja palvelua hyödyntävän käyttäjän täytyy tyytyä luonnollisella kielellä tehtyyn palvelun kuvaukseen [Bruijn *et al.*, 2005]. Näin ollen palveluiden liittäminen edellyttää aina aktiivista ihmisen toimintaa varmistamaan, että matalan tason tietorakenteita tulkitaan oikein ja että integrointi tapahtuu semanttisesti korrektisti [Bouras *et al.*, 2006]. Tällä perusteella perinteisten Web service -palveluiden heikkoutena järjestelmien integroinnin näkökulmasta voitaneen pitää palvelurajapintojen kuvauksiin ja palveluiden käyttämiin tietorakenteisiin liittyvän semanttisen merkityksen sekä palveluiden tarjoamien toimintojen kuvausten formalisoinnin puutetta.

6.3. Semanttinen Web service –teknologia ja tietojärjestelmien yhteistoiminnallisuus

Web service -teknologian käytöllä yritysten tietojärjestelmissä on tavoitteena mahdollistaa tietojärjestelmien integrointi. Perinteisten Web service -järjestelmien sujuvaa yhteentoimivuutta vaikeuttaa palveluiden ja toiminnallisuuden kuvaukseen liittyvän semantiikan muodollisen kuvauksen puuttuminen palvelun kuvauksesta. Ilman, että otetaan huomioon tietojärjestelmien semanttinen heterogeenisyys, ei voida saavuttaa tietojärjestelmien saumatonta yhteistoiminnallisuutta [Ushold and Gruninger, 2004]. Yrityksen tietojärjes-

telmiä kehitettäessä sovelluskehittäjien vaikeutena on monimutkaisia palveluja yhdistettäessä selvittää tietorakenteiden merkitys ja siten varmistaa semanttisesti korrekki integrointi [Bouras *et al.*, 2006]. Myös palvelun toiminnallisuuden suhteen on vaikeata varmistua, että tarjottu ja toivottu toiminnallisuuden kuvaus vastaavat toisiaan etenkin, jos sovellukset käyttävät eri termistöä [Bruijn *et al.*, 2005]. Kytkemällä semanttisen webin teknologioita yhteen palvelusuntautuneen arkkitehtuurin kanssa luomalla semanttisia Web service -palveluita uskotaan pystyttävän luomaan tie kohti (puoli)automaattista yritysten tietojärjestelmien integrointia [Bouras *et al.*, 2006]. Kun Web service teknologialla toteutettujen palveluiden kuvaukset ovat koneiden prosessoitavissa ja tulkittavassa, uskotaan pystyttävän automatisoituun Web service -palveluiden hyödyntämiseen niin organisaation sisäisesti kuin organisaatioiden välillä. Tämä käsittää palveluiden automatisoidun löytämisen, niiden käyttämisen, uusien toimintojen muodostamisen eri palveluita yhdistämällä ja sovellusten sujuvan yhteentoimivuuden.

6.3.1. WSDL ja ontologiat

W3C:n määrittelemässä Web Services -arkkitehtuurissa ei sinänsä määritellä palvelukuvaukseen liittyvän palvelun funktionaalisen kuvauksen muotoa [W3C, 2004d]. Arkkitehtuurin määrittelyssä todetaan funktionaalisen kuvauksen voivan olla esimerkiksi muutaman sana metadatan tai yhtäläillä esimerkiksi RDF tai OWL-S lausekkeita. Joten Web Services arkkitehtuuri mahdollistaa tietokoneiden tulkittavissa olevan semanttisen informaation lisäämisen palvelukuvauksiin. Vuonna 2007 W3C julkaisi suosituksen WSDL 2.0 -kielestä, jossa uutena piirteenä on mukana sovitus (mapping) RDF-kieleen. Suosituksessa WSDL-kielestä kuvataan kuinka WSDL rakenteet on mahdollista kuvata käyttäen RDF-kieltä. Koska RDF-kieli on tarkoitettu tietämyksen esittämiseen, voidaan sovittamalla WSDL rakenteet RDF-kieleen esittää formaali palvelukuvaus tietämyksen muodossa esittäen palvelukuvaus ontologiana. Sovitettaessa WSDL-rakenteet RDF-kieleen esitetään komponentteina määritellyt ominaisuudet (esimerkiksi kuvaus, rajapinta, sidonnat ja palvelu) RDF-kielessä resursseina ja niiden välisinä suhteina. Resurssit esitetään luokkina, jotka on määritelty OWL-kieltä käyttäen. [W3C, 2007]

6.3.2. Palveluiden kuvaus ontologioita käyttäen Web services -teknologiassa

Kehykeksi Web service -teknologialla toteutetun palvelun kuvauksesta ontologioita käyttäen on tehty useita eri ehdotuksia, kuten esimerkiksi The Ontology Web Language for Services (OWL-S) [Martin *et al.*, 2007] ja The Web Service

Modeling Ontology (WSMO) [Wang *et al.*, 2007]. Tavoitteena ehdotetuilla kehyksillä on esitellä yleinen formaali sanasto ontologian muodossa Web service-palveluiden kuvaamiseksi. Esittämällä palvelun kuvaus formaalisti ontologioiden muodossa uskotaan pystyttävän parantamaan automatisointia sopivan palvelun löytämiseen verkosta, sekä automatisoimaan valitun palvelun käyttöönotto. Esitetyissä malleissa esitetään ontologioiden avulla:

- Kuvaus palvelun toiminnallisuudesta, siis mitä palvelun avulla voidaan tehdä. Kuvauksen on tarkoitus palvella tarpeisiinsa sopivaa palvelua etsivää potentiaalista palvelun käyttäjää.
- Kuvaus palvelun toiminnasta, prosessista kuinka palvelu toimii. Monimutkaiset palvelut saattavat edellyttää useampaan otteeseen tapahtuvaa vuorovaikutusta palvelun tarjoajan ja käyttäjän välillä. Prosessin kuvauksessa esitetään kuinka prosessi etenee (palvelun kutsuminen, palvelun odottama syöte (input), odotettavissa oleva tulos (output), palvelun monitorointi ja virheistä toipuminen).
- Palvelun toteuttaman rajapinnan kuvaus.

Toistaiseksi ei ole kuitenkaan mitään mallia suositeltu tai standardoitu W3C:n toimesta. Toimintaperiaatteiltaan ja rakenteeltaan useimmat ehdotetuista malleista ovat hyvin lähellä toisiaan. Jos tarkastellaan esimerkiksi OWL-S ontologiaa ja WSMO ontologiaa, niin molemmissa kuvataan palvelun käyttämien informaation tilaa koskevat vaatimukset ennen ja jälkeen palvelun kutsusta, sekä mahdolliset palvelun käytön vaikutukset ulkopuolisille tekijöille. Palvelun vaikutuksia voi olla esimerkiksi luottokortin veloittaminen jne. Tosin monilla informaatiota tarjoavilla palveluilla ei ole muita seurauksia. Lisäksi, jos palvelu on muodostettu koostamalla toisista palveluista, voidaan kuvata prosessi kuinka palvelu käyttää toisia palveluita. Suurimpana erona esimerkiksi WSMO:n ja OWL-S:n välillä on, että WSMO:ssa käytetään välittäjiä (mediator) ratkaisemaan järjestelmien välisiä yhteensopivuusongelmia heterogeenisten järjestelmien välillä, kun taas OWL-S:n lähestymistavassa otaksutaan, että järjestelmien heterogeenisuus voidaan ratkaista palvelun toimesta formaalin kuvauksen ansiosta. WSMO:ssa käytettyjen välittäjien avulla ratkaistaan data, kommunikointi ja liiketoimintaprosessien -tasolla käytettävien sanastojen poikkeavuuksista aiheutuvat yhteensopimattomuudet [Bruijn *et al.*, 2005].

7. Liiketoimintaprosessien yhteistoiminnallisuus ontologioiden avulla

Visiossa semanttisesta webistä [Berners-Lee *et al.*, 2001] kuvataan nimenomaan eri yritysten tietojärjestelmien yhteistoiminnallisuutta, jossa liiketoimintatapahtumia suoritetaan yritysten välillä älykkäiden agenttien toimesta. Jotta älykkäiden agenttien on mahdollista hoitaa liiketoimintatapahtumia ihmisten sijaan, on agenttien ymmärrettävä yritysten liiketoimintaprosesseja ihmisten tapaan ja kyettävä suorittamaan liiketoimintatapahtumia ihmisten tapaan. Esimerkkejä liiketoimintatapahtumista, jotka voisivat soveltua myös agenttien hoidettavaksi, ovat tarjouspyynnöt, tarjoukset ja tilaukset. Tietojärjestelmien yhteistoiminnallisuudessa yritysten välillä on siis usein kyse elektronisen kaupan järjestelmien yhteistoiminnallisuudesta. Voitaneen ajatella, että pyrittäessä automatisoimaan yritysten välisiä liiketoimintatapahtumia kyseessä on liiketoimintaprosessien yhteensovittaminen yritysten välillä.

Pyrittäessä liiketoimintaprosessien yhteistoiminnallisuuteen usein ongelmana on kahden eri yrityksen liiketoimintaprosessien välinen yhteistoiminnallisuus. Jos jotain teknologiaa käyttämällä kyetään toteuttamaan organisaatioiden välisten liiketoimintaprosessien yhteistoiminnallisuus, niin todennäköisesti sama teknologia sopii käytettäväksi liiketoimintaprosessien yhteistoiminnallisuuteen organisaation sisäisten prosessien välillä. Liiketoimintaprosesseja mallinnetaan usein luomalla workflow-malli mallinnettavasta liiketoimintaprosessista. Johannesson *et al.* [2005] ovat tutkineet ontologioiden hyödyntämistä liiketoimintaprosessien kuvaamisessa ja heidän mukaansa liiketoimintaprosessin mallintamisessa tarvittavan tietämyksen esittäminen on mahdollista ontologian avulla. Pyrittäessä liiketoimintaprosessien yhteistoiminnallisuuteen eri yritysten välillä tulee osapuolten ymmärtää toistensa liiketoimintaprosessit. Tällöin vähimmäisvaatimuksena on ihmiseltä ihmiselle tapahtuvan tietämyksen siirto, ja kun halutaan automatisoida liiketoimintaprosessien yhteistoiminnallisuus edellytetään ihmiseltä tietokoneelle tapahtuvaa tietämyksen siirtämistä [Johannesson *et al.*, 2005].

Pyrittäessä liiketoimintaprosessien yhteistoiminnallisuuteen osapuolten tulee mallintaa prosessiensa vuorovaikutusta korkealla abstraktiotasolla liiketoimintatapahtumien näkökulmasta. Samanaikaisesti heidän tulee kytkeä sisäisiä toimintojaan ja resursseja liiketoimintasopimuksen mukaisessa yhteistoiminnassa hyödynnettäväksi sekä tarjota rajapinta ulkopuolisille yhteistoimintaa varten. Liiketoimintaprosessien menestyksellisen yhteistoiminnan edellytyksenä on, että osia järjestelmän sisäisistä toiminnoista paljastetaan ulko-

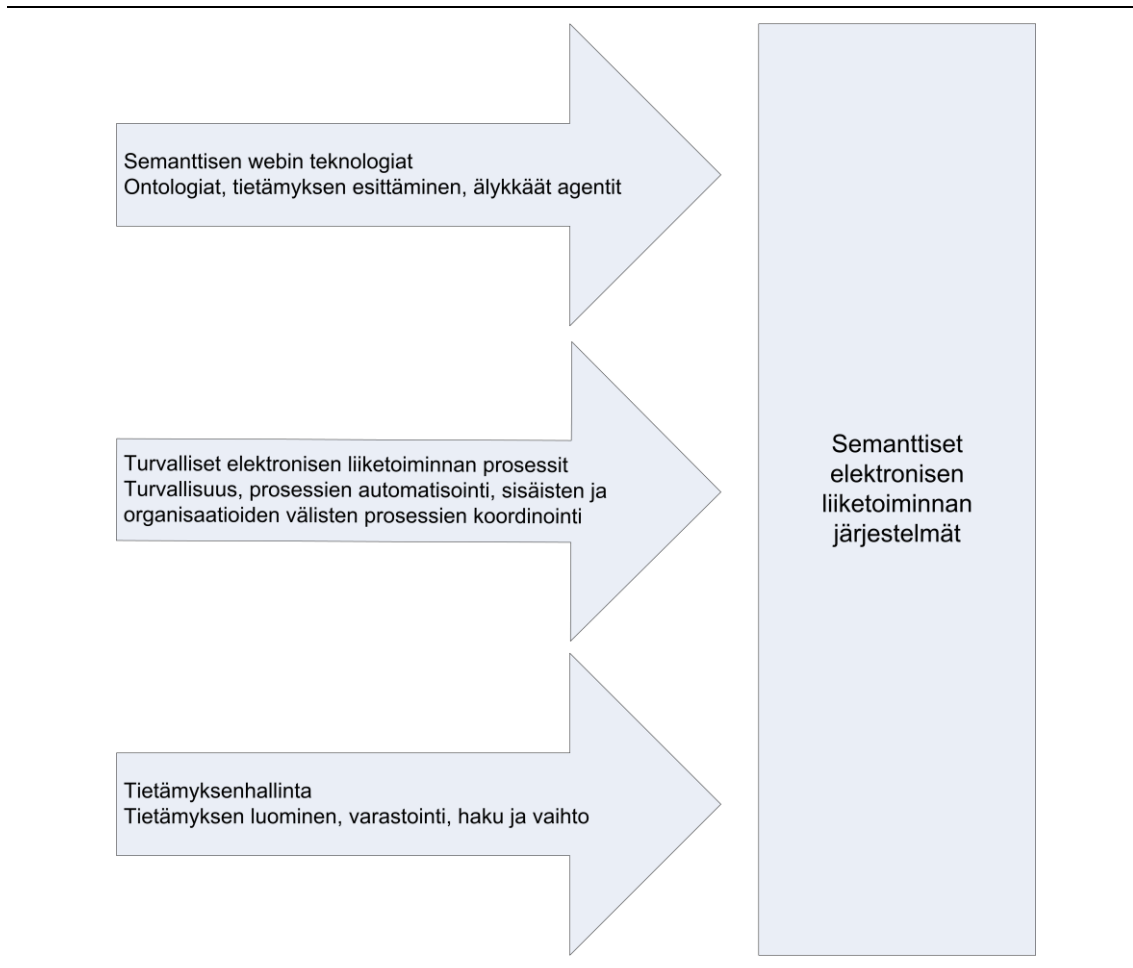
puolisille. [RIDE Consortium, 2008] Ulkopuolisille paljastettavien sisäisten toimintojen määrä luultavasti vaihtelee osapuolten välisen luottamuksen tason perusteella. Liiketoiminnan turvaamiseksi on sisäisten ja ulkoisten prosessien koordinoinnilla merkittävä rooli liiketoimintaprosessien yhteistoiminnallisuudessa.

7.1. Semantiikka elektronisen kaupan järjestelmissä

Elektroninen kauppa on yksi lupaavimmista viime vuosina liikemaailmalle esitellyistä informaatioteknologian sovelluksista ja sillä on suunnaton potentiaali kehittyä tuotteiden valmistuksen, vähittäiskaupan ja palveluiden operaatioissa [Durgin, 2008]. Elektronisen kaupan sovelluksilla on myös suuri vaikutus yritysten toimitusketjujen hallintaan. Mitä tahansa Internetissä tapahtuvaa kaupankäyntiin liittyvä tapahtumaa voidaan pitää elektronisena liiketoimintana [Durgin, 2008]. Singh *et al.* [2005, p.38] määrittelevät semanttisen elektronisen liiketoiminnan (Semantic E-business) seuraavasti:

Semanttinen elektroninen liiketoiminta on lähestymistapa tietämyksen hallitsemiseksi elektronisen liiketoiminnan prosessien koordinoinnissa hyödyntäen systemaattisesti semanttisen webin teknologioita.

Ottaen huomioon, että nykyinen Internet tulee kehittymään kohti semanttista webiä, niin myös elektronisen kaupan järjestelmissä tulee aktivoida tietorakenteiden ja datan semanttinen määrittely käyttäen XML-kieltä sekä XML:n käyttö datan vaihdossa [Durgin, 2008]. Semanttisen webin teknologiat ovat vielä kehitysvaiheessa, joten myös semanttisessa elektronisessa kaupassakin käytettävät teknologiat edellyttävät laajaa tutkimustyötä ennen kuin voidaan luoda käytännön sovelluksia laajassa mittakaavassa. Singh *et al.* [2005] esittävät vision semanttisesta elektronisesta liiketoiminnasta, jonka kehittyminen perustuu kolmeen eri aktiiviseen tutkimusaluseen. Heidän esittämän vision mukaan semanttisen elektronisen liiketoiminnan kehittyminen on riippuvainen semanttisen webin teknologioiden, tietämyksenhallinnan ja turvallisten elektronisen kaupan prosessien kehityksestä. Kuva 11 on esitetty näiden tutkimusalojen tärkeimpiä osa-alueita. Semanttisen webin teknologioista ontologioiden, tietämyksen esittämisen ja älykkäiden agenttien kehitys ovat tärkeimpiä tutkimusalueita semanttisen elektronisen kaupan kehittämisessä. Turvallisten elektronisen kaupan prosessien kehityksessä tärkeitä alueita ovat mm. prosessien automatisointi, yrityksen tietojärjestelmien integrointi ja asiankäsittelyn (workflow) ja aktiviteettien koordinointi yrityksessä ja yritysten välillä. Tietämyksenhallinnan tutkimuksen painoalueita on tietokoneiden tulkittavissa olevan tietämyksen vaihto ja päätöksenteon tukeminen.



Kuva 11 Visio semanttisesta elektronisesta liiketoiminnasta perustuu semanttisen webin teknologioihin, tietämyksen hallintaan ja elektronisen liiketoiminnan prosesseihin [Singh *et al.*, 2005]

Elektroninen liiketoiminta voi olla yritysten välistä (business-to-business (b2b)) tai yrityksen ja kuluttajien välistä (business-to-consumers (b2c)). Tavoitteena elektronisella kaupalla on luoda uusia ja laajentaa nykyisiä liiketoimintamalleja sekä leikata kustannuksia.

Internetissä tarjotaan asiakkaille muun muassa verkkokauppoja, verkossa toimivia huutokauppoja sekä hakupalveluita, joissa tarjotaan hintavertailuja eri kauppapaikkojen kesken. Perinteiseen kaupankäyntiin verrattuna verkkokaupan asiakas on uuden ongelman edessä. Asiakkaan löydettävä verkkokauppa, joka myy haluttua palvelua tai tuotetta sekä varmistauduttava, että tuotteen laatu, määrä, kaupankäynnin ehdot ja hinta vastaavat toivottua [Durgin, 2008]. Tuotteen löytämistä ja kauppojen vertailua helpottamaan on kehitetty hintavertailuagentteja (shopbots), jotka etsivät ja vertailevat tuotteita eri kauppapaikkojen välillä. Perinteisellä Internet-tekniikalla toteutetut kauppapaikat eroavat rakenteeltaan paljon toisistaan ja tuotteiden etsintä suoritetaan pää-

osin avainsanahakuna verkkokaupan sivuilta. Tällä on Durginin ja Sherifin [2008] mukaan kaksi vakavaa rajoitusta:

- Usein joudutaan laatimaan erikseen kääre (wrapper) tai rajapinta jokaista verkkokauppaa kohti, mikä lisäksi vaatii ylläpitoa.
- Avainsanoihin perustuva tuoteinformaation poiminta on rajoitettua ja johtaa helposti virheisiin.

Kuten yllä olevasta esimerkistä selvästi nähdään, on semanttisen tiedon ja sen analysoinnin puuttuminen selkeä puute nykyisellä tekniikalla toteutetussa sähköisessä kaupassa. Semanttisen sisällön lisääminen sähköisen kaupan sovelluksiin voi huomattavasti parantaa yhteistoiminnallisuutta sähköisen kaupan ja sen asiakkaana olevan älykkään agentin välillä.

7.2. Workflow-mallin luominen liiketoimintasopimuksesta sähköisessä sopimisessa ontologian avulla

Johannesson *et al.* [2005] esittelevät työssään järjestelmän, jossa liiketoimintasopimus esitetään käyttäen monitasoista sopimusontologiaa (Multi Tier Contract Ontology). Sopimusontologia tarjoaa tiedot liiketoimintasopimukseen liittyvistä velvollisuuksista ja osapuolilta odotettavista toiminnoista. Tätä edellä mainittua ontologiaa hyödynnetään luotaessa workflow-malli yritysten välisestä liiketoimintasopimuksesta ja voidaan näin saavuttaa palvelutason yhteistoiminnallisuus sähköisessä sopimisessa yritysten välillä. Heidän esittämällä toimintamallilla pyritään siirtymään perinteisestä informaatiotason yhteentoimivuuden tasolla toimivasta sopimushallintasoventuksesta järjestelmään, jossa yhteentoimivuuden taso on palvelutasolla. Sähköisen sopimisen toimiessa yhteentoimivuuden palvelutasolla voivat älykkäät agentit suorittaa sopimustarjousten tekemisen, neuvottelut sopimuksista ja näitä seuraavan sopimuksen toimenpanon seurannan.

Johannesson *et al.* [2005] hyödyntävät esittämässään järjestelmässä monitasoista liiketoimintasopimus-ontologiaa, joka on muodostunut kolmesta kerroksesta seuraavasti:

- Ylemmän tason yleinen sopimus-ontologia (Upper Level Core Contract Ontology), joka kuvaa sopimusten yleistä rakennetta ja soveltuu useimpien sopimustyyppien kuvaamiseen
- Eriomainen sovellusalakohtainen sopimus-ontologia (Specific Domain Level Contract Ontology), joka on kokoelma eri tyyppisten sopimusten kuvauksia. Jokainen sopimustyyppi on oma ontologiansa, kuten esimerkiksi huoneiston vuokrasopimus, myyntisopimus jne.

- Mallitason sopimus-ontologia (Template Level Contract Ontology), joka koostuu kokoelmasta yleisistä sopimusmalleista kuten esimerkiksi Kansainvälisen kauppakamarin (ICC) sopimusmalli kansainväliseen tavarakauppaan jne.

Johannesson *et al.*:n [2005] esittämä workflow-mallin luominen liiketoimintasopimuksesta perustuu ajatukseen, että luomalla tiivis sovitukset (mapping) luotavan workflow-mallin ja yritysten sisäisten prosessien välille, on mahdollista saavuttaa palvelutason yhteistoiminnallisuus yritysten tietojärjestelmien välille. Koska sopimuksen workflow-malli luodaan liiketoimintasopimukseen perustuen, sisältää workflow-malli kokonaisuudessaan sopimukseen sisältyvän liiketoimintalogiikan ja kaikki keskinäiset vuorovaikutustapahtumat. Näin ollen ontologia voi toimia tekniikkana, joka mahdollistaa tarvittavan tietämyksen välittämisen liiketoimintasopimukseen perustuvia liiketoimintatapahtumia suorittavien yritysten tietojärjestelmien välillä.

Workflow-mallin luomisella sopimusontologiaa käyttäen saattaa olla yritykselle muutakin hyötykäyttöä tietojärjestelmien yhteentoimivuuden lisäksi. Luotaessa workflow-malli liikesopimuksesta se on samalla myös korkean tason käsitteellinen malli, joka saattaa auttaa ylittä johtoa ja päätöksentekijöitä ymmärtämään vaatimukset sopimuksen noudattamiseen [Johannesson *et al.*, 2005]. Erityisesti tilanteissa, jossa sopimusehtoja on jollain tapaa loukattu, saattaa käsitteellinen malli olla hyödyllinen selvittämään mahdollisia eri toimintavaihtoehtoja ja auttamaan arvioimaan seuraamuksia joita sopimuksen noudattamatta jättämisestä saattaa olla.

7.3. Liiketoimintaprosessin monitorointi

Yrityksen tuottavuus ja kannattavuus on riippuvainen sen liiketoimintaprosessien tehokkuudesta. Liiketoimintaprosessien etenemisen ja tehokkuuden seuraaminen on tärkeä informaation lähde yrityksen johtamiseen liittyvässä päätöksenteossa. Yksi useimpien liikeyritysten ydintoiminnoista on kaupankäynti toisten yritysten ja organisaatioiden kanssa. Liiketoimintaprosessit organisaatioiden välillä tapahtuvat yleensä organisaatioiden välisten sopimusten perusteella ja sopimukset ovat yleensä laadittu siten, että sopimusten noudattaminen on eduksi molemmille osapuolille jolloin on sopimusosapuolten edunmukaista liiketoimintaprosessien optimointi parhaan mahdollisen tehokkuuden saavuttamiseksi ja siten voiton maksimointi [Kabilan *et al.*, 2003]. Joten yritysten välisten prosessien yhteensovittamisessa ei riitä, että saavutetaan yhteistoiminnallisuus prosessien muodostavien tapahtumien osalta, vaan myös prosessien etenemistä ja prosessit muodostavien tapahtumien suorittamista tulee pystyä seuraamaan.

Yritysten tietojärjestelmien yhteistoiminnallisuudessa älykkäiden agenttien on mahdollista hyödyntää eri tietojärjestelmiä monimutkaisten liiketoimintaprosessien suorittamisessa. Koska prosessien läpivienti tapahtuu automatisoidusti eri järjestelmien välillä, tulisi myös prosessien seuranta ja analysointi pystyä automatisoimaan. Manoj *et al.* [2005] esittelevät semanttisen webin teknologioihin perustuvan elektronisen liiketoiminnan järjestelmän, jossa älykkäät agentit suorittavat prosessien valvonnan ja tapahtumien seuraamisen. Heidän järjestelmässään, jossa prosessien suorittaminen tapahtuu hyödyntämällä sidottuja sovelluksia prosesseissa tarvittavien tehtävien suorittamiseen, suoritetaan prosessien valvonta ohjelmistoagenttien toimesta liiketoimintaprosessien suorituksen lomassa.

Prosessien mallintamiseen käytettyjen kielten avulla voidaan esittää workflow-mallin aktiviteetit, aktiviteettien eri tilat, tapahtumiin liittyvät tilojen muutokset ja aktiviteetteihin liittyviä poikkeuksia. Prosessien mallintamiskielten avulla ei pystytä kuitenkaan ilmaisemaan semantiikkaa, joka vaadittaisiin kuvaamaan aktiviteetteja ja niiden suorittamista kuvaavia kriteereitä siten, että voitaisiin yleisellä tasolla voitaisiin päätellä aktiviteettien ja koko prosessin suorittamista [Thomas *et al.*, 2005].

Sekä Manoj *et al.* [2005] että Kabilan *et al.* [2003] ehdottavat ontologioiden käyttöä liiketoimintaprosessin suorituksen kuvaamiseen. Manoj *et al.* [2005] tarkastelevat työssään Web service -teknologiaan perustuvassa järjestelmässä suoritettavia prosesseja ja Kabilan *et al.* [2003] kuvaavat organisaatioiden väliseen sopimukseen perustuvan liiketoimintaprosessin suorituksen seuraamista. Ontologioita käyttäen on mahdollista luoda semanttinen kuvaus liiketoimintaprosessiin kuuluvien aktiviteettien suorittamiseen liittyvistä vaatimuksista yhdessä aktiviteettien kuvauksen kanssa, mitä liiketoimintaprosessia kokonaisuudessaan valvovat agentit voivat hyödyntää keräten tietoa useita eri kriteereitä käyttäen prosessin suoritusta analysoidessaan [Thomas *et al.*, 2005]. Ontologioiden käyttö yhdessä prosessin mallinnuskielen kanssa mahdollistaa rakenteisen tietämyksen esittämisen liiketoimintaprosessista ja mahdollistaa rakenteen abstraktoinin säilyttämisen alla olevan prosessin suhteen. Rakenteen abstraktoinnin ansiosta ei ole tarpeen luoda tiivistä sovitusta prosessin suoritukseen kuuluvien aktiviteettien ja niiden tapahtumia seuraavien agenttien välille.

8. Yhteenveto

Tietojärjestelmien yhteistoiminnallisuus tarkoittaa kykyä siirtää tietoa sujuvasti eri järjestelmien välillä. Siirrettäessä tietoa järjestelmien välillä voidaan ontologioita hyödyntää varmistettaessa, että kaikki tietoa käsittelevät järjestelmät käyttävät tietoa samassa semanttisessa merkityksessä. Ontologia toimii tällöin järjestelmien välisenä, niille yhteisenä kielenä asioiden merkityksen määrittelyyn.

Tietojärjestelmien yhteydessä ontologialla tarkoitetaan sovellusalueen tietämystä kuvaavaa rakenteellista sanastoa, joka koostuu luokista, niiden välisistä suhteista ja ominaisuuksista. Ontologia määrittelee kohdealueella käytettävän sanaston, termien merkityksen ja käsitteiden väliset suhteet käyttäen muodollisia sääntöjä. Sanasto koostuu yleensä kohdemaailman käsitteistöstä ja näiden käsitteiden väliset suhteet määritellään hierarkkisesti. Määrittelemällä ontologiassa käytetyt käsitteet attribuutteineen, niiden väliset suhteet ja rajoitukset sekä määrittelemällä käsitteiden esiintyminä esiintyvät oliot pystytään välittämään tietoa järjestelmien välillä siten, että järjestelmät pystyvät ymmärtämään tiedon aiotun merkityksen loogisten sääntöjen avulla. Ontologia tarjoaa tietojärjestelmien yhteistoimintaan osallistuville tietojärjestelmille yhteisen tavan välittää kohdealueeseen liittyvää tietämystä tietojärjestelmien välillä. Olennaista on, että ontologioilla ei välitetä viestejä järjestelmien välillä, vaan niillä kuvataan tietämystä sovellusalueelta.

Avaimena yhteistoiminnallisuuteen on ontologioiden jakaminen yhteistoimintaan osallistuvien järjestelmien kesken, jolloin järjestelmät jakavat saman käsityksen sovellusalueesta ja sen kuvauksesta. Ennen kuin on mielekästä hyödyntää ontologioita yhteistoiminnallisuuteen semanttisella tasolla, on tietojärjestelmien pystyttävä yhteistoimintaan teknisellä tasolla sekä omattava syntaktisella ja rakenteellisella tasolla toimiva yhteistoiminnallisuus viestien välittämiseen järjestelmien välillä.

Ontologioiden roolina on mahdollistaa semanttisen tietämyksen liittäminen järjestelmien välillä siirrettävään informaatioon siten, että voidaan varmistua yhteistoimintaan osallistuvien järjestelmien käyttävän termejä samassa semanttisessa merkityksessä. Koska XML-kieltä käyttäen voidaan toteuttaa tietojärjestelmien yhteistoiminnallisuus syntaktisella ja rakenteellisella tasolla, tarvitaan semanttisen tason yhteistoiminnallisuuteen muita tekniikoita. Ontologioiden esittämiseksi tietojärjestelmissä on kehitetty XML-kieleen perustuvia kuvauskieliä.

Ontologioita luodaan käyttötarkoituksesta riippuen erilaisilla erikoistamisen tasoilla ja eri sovellusaloille. Käytännössä on hylätty ajatus luoda kaiken kattava yksi ainoa ontologia sen luomisen vaikeudesta johtuen. Sen sijaan pyritään luomaan ontologioita käyttötarpeen mukaan, jotka ovat juuri riittävän laajoja kohdealueiden kuvaamiseksi. Yksi tapa luoda ontologioita on koostaa ontologioita uuteen käyttötarkoitukseen aiemmin eri käyttötarkoituksiin luoduista ontologioista. Käyttämällä uudelleen aiemmin luotuja ontologioita on mahdollista saavuttaa säästöjä ontologioiden muodostamiseen tarvittavan työmäärän pienentyessä. Järjestelmien käyttäessä eri ontologioita on olennaista, että käytetyt ontologiat ovat vertailukelpoisia keskenään, kun tarkoituksena on hyödyntää ontologioita tietojärjestelmien yhteistoiminnallisuuteen. Ontologiat perustuvat saman termistön käyttöön ja ovat siten vertailukelpoisia keskenään, kun ne muodostetaan erikoistamalla samaa ylemmän tason ontologiaa.

Tietojärjestelmien yhteistoimintaan osallistuvilla tahoilla on usein eriäviä näkemyksiä kohdealueen kuvaamisessa käytettävistä termeistä ja niiden välisistä suhteista. Yhteistoiminnallisuuteen käytettyjen jaettujen ontologioiden standardoinnilla on suuri merkitys yhteisen näkemyksen saavuttamisessa ja yhteistoiminnallisuuden mahdollistamisessa. Oman vaikeutensa ontologioiden muodostamiseen tuo nopea muutostahti sovellusalueella. Ontologian kuvaaman kohdealueen muuttuminen edellyttää muutosten tekemistä myös ontologiaan. Koska ontologiat usein käyttävät uudelleen ja laajentavat toisia ontologioita, heijastuvat muutokset muutoksen kohteena olevaa ontologiaa käyttäviin ontologioihin ja sovelluksiin. Tästä johtuen saattaa ontologioiden evoluution ja versioiden hallinta olla ongelmallista.

Ontologiat määrittelevät tietojärjestelmien käyttöön sovellusalueella käytettävän termistön ja ontologioiden avulla on mahdollista varmistaa, että eri järjestelmät käyttävät termejä samassa merkityksessä, mikä tukee tietojärjestelmien yhteistoiminnallisuutta. Tavoitteena ontologioiden käytöllä tietojärjestelmien välisessä kommunikoinnissa on automatisoida tietojärjestelmien yhteistoiminnallisuus. Kyettäessä automatisoimaan tietojärjestelmien yhteistoiminnallisuus on mahdollista saavuttaa huomattavia taloudellisia säästöjä tietojärjestelmien kehittämisessä tarvittavan työmäärän pienentyessä. Myös tiedon laatua on mahdollista parantaa, kun eri lähteistä tulevan datan yhtenäisyys voidaan tarkastaa automatisoidusti tietokoneiden pystyessä analysoimaan tiedon semanttista sisältöä. Ontologioiden avulla on myös mahdollista kuvata sovellusalueen tietämystä muodossa, jota on mahdollista tulkita tietokoneiden avulla. Sovellusalueen tietämyksen esittämisestä ontologian avulla voidaan hyötyä ontologioiden mahdollistaessa tietämyksen uusiokäyttöä, tiedon ylläpidon helpottumisella ja tietämyksen kehittymisellä pitkällä aikavälillä.

Tämän hetkinen trendi yritysten tietojärjestelmiä kehitettäessä on Web service -teknologian käyttö. Web services -teknologia perustuu palveluarkkitehtuurin käyttöön, missä uusia sovelluksia voidaan luoda yhdistelemällä sovelluksessa tarvittavan toiminnallisuuden tarjoavia palveluita. Perinteisten Web services -sovellusten palvelukuvaukset eivät sisällä semanttista kuvausta palvelusta, jonka seurauksena palvelujen käyttöönotto tapahtuu manuaalisesti ihmisvoimin. Lisäämällä semanttinen kuvaus palvelukuvaukseen on mahdollista automatisoida uusien palvelujen käyttöönottoa ja mahdollisesti uusien sovellusten luomista palveluita yhdistämällä.

Yritysten liiketoimintaprosessit on mahdollista suorittaa yhteistoiminnallisissa järjestelmissä automaattisesti eri järjestelmien välillä älykkäiden agenttien toimesta. Koska liiketoimintaprosessien seuraaminen on usein elintärkeää yrityksen menestykselle toiminnalle, menestykselle tietojärjestelmien yhteistoiminnalle on edellytyksenä automatisoitu prosessien monitorointi. Agenttien dynaamisesti suorittama prosessien toimien seuranta on mahdollista käyttämällä ontologioita prosessien suorituksen kuvaamiseen.

Tietojärjestelmien yhteistoiminnallisuuteen pyrittäessä ontologioita tulisi hyödyntää mekanismina, jonka avulla välitetään järjestelmien välillä siirrettävän tiedon semanttinen sisältö. Jakamalla yleisesti hyväksytyjä ontologioita tietojärjestelmien kesken luodaan edellytykset tietojärjestelmien yhteistoiminnallisuuteen. Edellytyksenä ontologioiden hyödyntämiselle yhteistoiminnallisuuteen on järjestelmille yhteisen kielen käyttö ontologioiden määrittelyyn, mihin XML-kieleen perustuvien standardoitujen kuvauskielten käyttö on yleisesti hyväksytty ratkaisu.

Ontologioiden käytöllä tietojärjestelmien yhteydessä näyttää olevan lupaavia mahdollisuuksia tietojärjestelmien yhteistoiminnallisuuteen pyrittäessä ja ontologioiden hyödyntämiseen liittyen tehdään paljon tutkimustyötä. Ontologioiden käyttö on kuitenkin toistaiseksi vähäistä johtuen todennäköisesti siitä, että ontologioita hyödyntävä teknologia on edelleen kehitysvaiheessa. Luultavasti edellytetään ontologioita hyödyntävien teknologioiden kypsymistä ennen kuin yritykset alkavat hyödyntämään ontologioita tietojärjestelmissä suuremmassa mittakaavassa. Lisäksi ontologioiden hyödyntäminen tietojärjestelmien yhteistoiminnallisuuteen eri organisaatioiden välillä edellyttää yleisesti hyväksytyjen ontologioiden luomista eri sovellusalueille. Myös tällä saralla on tehtävänä paljon standardointi- ja kehitystyötä ennen kuin ontologioiden laajamittainen hyödyntäminen on mahdollista.

Tässä opinnäytetyössä on analysoitu teoreettisia tutkimuksia ontologioiden käytöstä tietojärjestelmien yhteistoiminnallisuuteen ja pyritty muodostamaan kokonaiskuva teoreettista pohjaa ontologioiden käytöstä tietojärjestelmien yh-

teistoiminnallisuuteen tutkineiden tutkijoiden tuloksista. Jatkossa olisi tarpeen testata teorian toimivuutta käytännössä rakentamalla reaali maailman järjestelmiä ja tutkimalla niiden toimintaa.

Viitteet :

- [Aparicio *et al.*, 2005] Aparicio, A.S., Farias, O.L.M. & Santos, N.d. 2005, "Applying ontologies in the integration of heterogeneous relational databases", *Proceedings of the 2005 Australasian Ontology Workshop - Volume 58* Australian Computer Society, Inc., Sydney, Australia, pp. 11-16.
- [Berners-Lee *et al.*, 2001] Berners-Lee, T., Hendler, J. & Lassila, O., 2001, "The Semantic Web", *Scientific American*, vol. 284, no. 5, pp. 34-43.
- [Bouras *et al.*, 2006] Bouras, A., Gouvas, P. & Mentzas, G., 2006, "A Semantic Service-Oriented Architecture for Business Process Fusion" in *Semantic Web Technologies and E-Business: Towards the Integrated Virtual Organization and Business Process Automation*, ed. A.F. Salam, Idea Group Publishing, Hershey, PA, USA, pp. 40-76.
- [Brandt *et al.*, 2008] Brandt, S.C., Morbach, J., Miatidis, M., Theissen, M., Jarke, M. & Marquardt, W., 2008, "An ontology-based approach to knowledge management in design processes", *Computers & Chemical Engineering*, vol. 32, no. 1, pp. 320-342.
- [Bruijn *et al.*, 2005] Bruijn, J.d., Fensel, D., Keller, U. & Lara, R., 2005, "Using the web service modeling ontology to enable semantic e-business", *Communications of the ACM. Vol.48*, vol. 48, no. 12, pp. 43-47.
- [Chandrasekaran *et al.*, 1999] Chandrasekaran, B., Benjamins, V.R. & Josephson, J.R., 1999, "What Are Ontologies, and Why Do We Need Them?", *IEEE Expert Intelligent Systems & Their Applications*, vol. 14, no. 1, pp. 20-26.
- [Dou and LePendu, 2006] Dou, D. & LePendu, P. 2006, "Ontology-based integration for relational databases", *Proceedings of the 2006 ACM symposium on Applied computing* ACM, Dijon, France, pp. 461-466.
- [Durgin, 2008] Durgin, J.K., 2008, "The semantic web: a catalyst for future e-business", *Kybernetes*, vol. 37, no. 1, pp. 49-65.
- [Gali *et al.*, 2004] Gali, A., Chen, C.X., Claypool, K.T. & Uceda-Sosa, R., 2004, "From Ontology to Relational Databases" in *Conceptual Modeling for Advanced Application Domains*, eds. S. Wang, K. Tanaka, S. Zhou, et al, Springer Berlin / Heidelberg, Berlin, pp. 278-289.
- [Geraci *et al.* , 1991] Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H. & Springsteel, F. 1991, *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*, The Institute of Electrical and Electronics Engineers Inc.

- [Gruber, 2007] Gruber, T., 2007, "Ontology of Folksonomy: A Mash-Up of Apples and Oranges", *International Journal on Semantic Web and Information Systems*, vol. 3, no. 1, pp. 1-11.
- [Gruber, 1995] Gruber, T.R., 1995, "Toward principles for the design of ontologies used for knowledge sharing?", *International Journal of Human-Computer Studies*, vol. 43, no. 5-6, pp. 907-928.
- [Gruber, 1993] Gruber, T.R., 1993, "A translation approach to portable ontology specifications", *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220.
- [Gruber, 2008] Gruber, T., 2008, "Ontology" in *Encyclopedia of Database Systems*, eds. L. Liu & M.T. Özsu, Springer-Verlag, Berlin.
- [Guarino, 1998] Guarino, N., 1998, "Formal Ontology in Information Systems" in *Formal Ontology in Information Systems. Proceedings of FOIS'98*, ed. N. Guarino, IOS Press, Amsterdam, pp. 3-15.
- [Guarino, 1995a] Guarino, N., 1995a, "Formal ontology, conceptual analysis and knowledge representation", *International Journal of Human-Computer Studies*, vol. 43, no. 5-6, pp. 625-6.
- [Guarino, 1995b] Guarino, N., 1995b, "Ontologies and Knowledge Bases: Towards a Terminological Clarification" in *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, ed. N. Mars, IOS Press, Amsterdam, pp. 25-32.
- [Guarino, 1997] Guarino, N., 1997, "Semantic matching: Formal ontological distinctions for information organization, extraction, and integration" in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, ed. M.T. Pazienza, Springer Berlin / Heidelberg, Berlin, pp. 139-170.
- [Hevner *et al.*, 2004] Hevner, A.R., March, S.T. & Park, J., 2004, "Design Science in information systems research", *MIS Quarterly*, vol. 28, no. 1, pp. 75-105.
- [Hübner *et al.*, 2001] Hübner, S., Neumann, H., Schuster, G., Stuckenschmidt, H., Wache, H. & Vögele, T. 2001, "Ontology-based integration of information - a survey of existing approaches", *Proceedings of the IJCAI-01 Workshop on Ontologies and Information, International Joint Conference on Artificial Intelligence*, eds. A. GómezPérez, M. Gruninger, H. Stuckenschmidt & M. Uschold.
- [Johannesson *et al.*, 2005] Johannesson, P., Kabilan, V. & Zdravkovic, J. 2005, "Using Multi Tier Contract Ontology to Deduce Contract Workflow Models for Enterprise Process Interoperability", *Proceedings of 2nd INTERPEMOI open workshop on Enterprise Models and Interoperability colocated with CAISE 2005, Porto*.

- [Kabilan *et al.*, 2003] Kabilan, V., Johannesson, P. & Rugaimukamu, D., 2003, "Business Contract Obligation Monitoring through Use of Multi Tier Contract Ontology" in *OTM Workshops 2003, LNCS 2889*, eds. R. Meersman & Z. Tari, Springer-Verlag, Berlin, pp. 690-702.
- [Martin *et al.*, 2007] Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., Mcguinness, D.L., Sirin, E. & Srinivasan, N., 2007, "Bringing Semantics to Web Services with OWL-S", *World Wide Web*, vol. 10, no. 3, pp. 243-277.
- [Noy and Klein, 2004] Noy, N.F. & Klein, M., 2004, "Ontology Evolution: Not the Same as Schema Evolution", *Knowledge and Information Systems*, vol. 6, no. 4, pp. 428-440.
- [Pollock, 2001] Pollock, J.T., 2001, *The Big Issue: Interoperability vs. Integration* International Organization for Standardization [Online]. Saatavilla: http://isotc.iso.org/livelink/livelink/fetch/-1287650/2426503/XML_semantic_interoperability-Integration_Pollock_1.pdf?nodeid=2425856&vernum=0 (luettu 6.4.2008).
- [Rahm and Bernstein, 2001] Rahm, E. & Bernstein, P.A., 2001, "A survey of approaches to automatic schema matching", *The VLDB Journal*, vol. 10, no. 4, pp. 334-350.
- [RIDE Consortium, 2008] RIDE Consortium, 2008, *A Roadmap for Interoperability of eHealth Systems in Support of COM 356 with Special Emphasis on Semantic Interoperability* European Institute for Health Records [Online]. Saatavilla: <http://www.srdc.metu.edu.tr/webpage/projects/ride/deliverables/D4.4.1RIDE-RoadmapIII-v1.1.doc> (luettu 29.4.2008).
- [Singh *et al.*, 2005] Singh, R., Iyer, L.S. & Salam, A.F., 2005, "The Semantic E-Business Vision", *Communications of the ACM*, vol. 48, no. 12, pp. 38-41.
- [Smith, 2003] Smith, B., 2003, "Ontology" in *Blackwell Guide to the Philosophy of Computing and Information*, ed. L. Floridi, Blackwell, Oxford, pp. 155-166.
- [Smith, 1995] Smith, B., 1995, "Formal ontology, common sense and cognitive science", *International Journal of Human-Computer Studies*, vol. 43, no. 5-6, pp. 641-667.
- [Stuckenschmidt *et al.*, 2000] Stuckenschmidt, H., Wache, H., Vögele, T. & Visser, U. 2000, "Enabling technologies for interoperability", *Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, eds. U. Visser & H. Pundt, University of Bremen, Bonn, pp. 35-47.

- [Terzi *et al.*, 2003] Terzi, E., Vakali, A. & Hacid, M. 2003, "Knowledge Representation, Ontologies, and the Semantic Web", *Web Technologies and Applications: 5th Asia-Pacific Web Conference, APWeb 2003*, eds. M.E. Orłowska, Y. Zhang & Z. Zhou, Springer Berlin / Heidelberg, New York, pp. 594-599.
- [Thomas *et al.*, 2005] Thomas, M., Redmond, R., Yoon, V. & Singh, R., 2005, "A semantic approach to monitor business process", *Communications of the ACM*, vol. 48, no. 12, pp. 55-59.
- [Uschold and Gruninger, 1996] Uschold, M. & Gruninger, M., 1996, "Ontologies: principles, methods, and applications", *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93-155.
- [Uschold and Gruninger, 2004] Uschold, M. & Gruninger, M., 2004, "Ontologies and semantics for seamless connectivity", *SIGMOD Rec.*, vol. 33, no. 4, pp. 58-64.
- [W3C, 2007] W3C, 2007, *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer* W3C [Online]. Saatavilla: <http://www.w3.org/TR/2007/REC-wsd120-primer-20070626/#WSDL-PART1> (luettu 8.4.2008).
- [W3C, 2006] W3C, 2006, *Extensible Markup Language (XML) 1.0 (Fourth Edition)* W3C [Online]. Saatavilla: <http://www.w3.org/TR/2006/REC-xml-20060816/> (luettu 1.4.2008).
- [W3C, 2004a] W3C, 2004a, *OWL Web Ontology Language Semantics and Abstract Syntax* W3C [Online]. Saatavilla: <http://www.w3.org/TR/2004/REC-owl-semantics-20040210> (luettu 18.12.2007).
- [W3C, 2004b] W3C, 2004b, *OWL Web Ontology Language Guide* W3C [Online]. Saatavilla: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> (luettu 1.4.2008).
- [W3C, 2004c] W3C, 2004c, *RDF Primer* W3C [Online]. Saatavilla: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> (luettu 15.1.2008).
- [W3C, 2004d] W3C, 2004d, *Web Services Architecture* W3C [Online]. Saatavilla: <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/> (luettu 8.4.2008).
- [Wang *et al.*, 2007] Wang, H.H., Gibbins, N., Payne, T., Saleh, A. & Sun, J. 2007, "A Formal Semantic Model of the Semantic Web Service Ontology (WSMO)", *ICECCS '07: Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007)* IEEE Computer Society, Washington, DC, USA, pp. 74-83.