Tietojenkäsittelytiede **23** Kesäkuu 2005 sivut 60–70
Toimittaja: Jorma Tarhio

Agenttipohjaiset liikkuvan tietojenkäsittelyn sovellukset FIPA-arkkitehtuurissa

Heikki Helin
TeliaSonera Finland
heikki.j.helin@teliasonera.com

Tiivistelmä

Langattoman tietoliikenteen päämääränä on mahdollistaa tietojenkäsittely missä tahansa ja milloin tahansa. Tällaiset ympäristöt kuitenkin asettavat huomattavia vaatimuksia sovellutuksille. Tässä artikkelissa esittelen, miten ohjelmistoagenttiteknologian avulla voitaisiin liikkuvuuden ja langattomuuden sovelluksille aiheuttamia ongelmia ainakin osittain ratkaista. Ohjelmistoagenttiteknologia on suhteellisen uusi suunnittelu- ja ohjelmointiparadigma, jolla pyritään ratkaisemaan monimutkaisten hajautettujen järjestelmien suunnittelu- ja toteutusongelmia.

1 Johdanto

Langaton tietoliikenne tulee olemaan oleellinen osa tietojenkäsittelyä tulevaisuudessa. Monilla aloilla liikkuvuus on välttämätöntä jo nykyisin, ja teknologioiden kehittyessä langaton tietoliikenne leviää myös aloille, joilla se ei ole vielä käytössä. Langattoman tietoliikenteen ja kannettavien päätelaitteiden kehittyminen mahdollistaa erilaisten kiinteän verkon palveluiden käyttämisen missä tahansa ja milloin tahansa käyttäen erilaisia päätelaitteita. Tällainen ympäristö kuitenkin asettaa huomattavia vaatimuksia arkkitehtuureille, joita käyttäen langattomassa ympäristössä toimivia sovelluksia voidaan kehittää. Langaton ympäristö eroaa kiinteän verkon ympäristöstä kahdella olennaisella tavalla. Ensinnäkin, käyttäjä toimii ympäristössä, jossa voi samanaikaisesti olla useita erilaisia ominaisuuksiltaan suuresti eroavia langattomia verkkoja. Tästä ja langattomien verkkojen ominaisuuksista kuten vaatimattomasta siirtonopeudesta ja pitkistä ja vaihtelevista viipeistä seuraa, että verkon tarjoama laatutaso (Quality-of-Service, QoS) voi muuttua paljon riippuen käytössä olevasta verkkoteknologiasta. Perinteisesti sovellukset on suunniteltu toimimaan nopeassa ja luotettavassa tietoliikenneverkossa. Tällaiset sovellukset toimivat huonosti tai eivät lainkaan, jos käytettävä tietoliikenneverkko ei täytä näitä vaatimuksia. Toiseksi, käyttäjällä voi olla käytössään useita erilaisia päätelaitteita. Erilaiset päätelaitteet asettavat erilaisia vaatimuksia käytettäville sovelluksille ja palveluille, joita käyttäjän on mahdollista käyttää. Jotta laitteen kuljettaminen mukana olisi vaivatonta, on laitteen fyysisen koon oltava pieni, jolloin niiden teknisistä ominaisuuksista on tingittävä.

Monimutkaisten hajautettujen järjestelmien suunnittelu on tunnetusti vai-

OHJE KIRJAPAINOLLE: B5-arkin vasen- ja yläreuna kohdistetaan A4-arkin vasempaan ja yläreunaan. Näin pitäisi marginaaliksi tulla taitteen puolella noin 33 mm ja muualla noin 22 mm.

keaa. Tällaisten sovellusten toteuttaminen langattoman tietoliikenteen ympäristöön on vielä vaikeampaa, koska sovellusten on otettava huomioon ympäristön aiheuttavat ongelmat muiden hajautukseen liittyvien ongelmien lisäksi. Ohjelmistoagenttiteknologian uskotaan olevan tehokas suunnittelu- ja toteutusmenetelmä monimutkaisille hajautetuille järjestelmille. Ohjelmistoagenttiteknologia on saanut huomattavasti huomiota erityisesti 90-luvun lopulla sekä tutkimuspiireissä että kaupallisessa maailmassa. Vaikka ohjelmistoagenttiteknologiaan kohdistunut huomio on laantunut, tätä teknologiaa tutkitaan maailmalla edelleen hyvin aktiivisesti. Ohjelmistoagenttiteknologian ohella toinen alaa mullistava kehitysalue on semanttinen web [3]. Semanttisen webin avulla verkossa oleva tieto voidaan esittää muodossa, jonka erilaiset ohjelmistot – vaikkapa ohjelmistoagentit – voivat ”ymmärtää” tiedon semantiikan ilman, että se ohjelmoidaan eksplisiittisesti mukaan ohjelmistoon. Tämä puolestaan mahdollistaa automaattisen päättelyn käsillä olevasta tiedosta. Ohjelmistoagenttien, semanttisen webin, ja langattoman tietoliikenteen yhdistäminen muodostaa mielenkiintoisen ja hedelmällisen tutkimusalueen. Yhdistelmä mahdollistaa uudenlaisen ympäristön sovellusten suunnitteluun ja toteuttamiseen ottaen huomioon langattoman tietoliikenteen asettamat vaatimukset.

Tässä artikkelissa esittelen agenttipohjaisen viitearkkitehtuurin sovellusten kehittämiseen langattoman tietoliikenteen ympäristöön. Arkkitehtuuri koostuu useista erillisistä komponenteista, jotka osaltaan mahdollistavat agenttien tehokkaan toiminnan langattoman tietoliikenteen ympäristöissä ja siten tarjoavat alustan sovellusten kehittämiseen. Ensimmäinen komponentti on langattomaan tietoliikenteeseen liittyvät ontologiat, jotka muiden ontologioiden kanssa yhdessä mahdollistavat ohjelmistoagenttipohjaisten sovellusten tehokkaan toiminnan langattomissa ympäristöissä. Esittelen muutamia tällaisia ontologioita luvussa 2. Toinen komponentti on abstrakti viitearkkitehtuuri, joka osaltaan mahdollistaa erilaisille alustoille kehitettyjen järjestelmien yhteensopivuuden. Viitearkkitehtuuri esitellään tarkemmin luvussa 3. Kolmas yhteensopivuuden mahdollistaja on ohjelmistoagenttien standardoitu vuorovaikutus, joka on sovitettu langattoman tietoliikenteen vaatimuksiin (luku 4). Kaikki nämä komponentit on osittain standardoidu FIPA:ssa (Foundation for Intelligent Physical Agents [13]) ja toteutettu esimerkiksi Jadessa [2], joka on yleisimmin käytössä oleva FIPA-yhteensopiva agenttialusta.

2 Langattoman ympäristön ontologiat

Ontologioiden avulla voidaan kuvata datan semantiikka eksplisiittisesti, joka puolestaan mahdollistaa tietämyksen jakamisen ja uudelleenkäytön [15]. Tietojenkäsittely-ympäristössä ontologiat mahdollistavat koneiden tai ohjelmistojen välisen vuorovaikutuksen siten, että osapuolet voivat olla varmoja, että he keskustelevat samasta asiasta ja ”ymmärtävät” siten toisiaan paremmin. Käytännössä ontologioilla kuvataan käsitteitä ja niiden välisiä suhteita formaalisti siten, että esimerkiksi ohjelmistoagentit voivat niitä käsitellä. Ontologia kuvataan ontologiakielellä, joista käytetyimpiä nykyään ovat RDF(S) [4] ja OWL [1]. Ontologia sisältää luokkia (käsitteitä) ja niiden välisiä ominaisuuksia. Vapaasti määriteltävien ominaisuuksien lisäksi RDF(S) ja OWL sisältävät joitakin ”sisäänrakennet-

tuja" ominaisuuksia, kuten subclassOf-relaation, jonka avulla voidaan rakentaa taksonomioita eli asioiden hierarkiasuhteita. Täten ontologia voidaan käsittää tietomalliksi. Ontologian luokista voidaan edelleen luoda ilmentymiä eli instansseja, jotka edustavat kyseistä luokkaa.

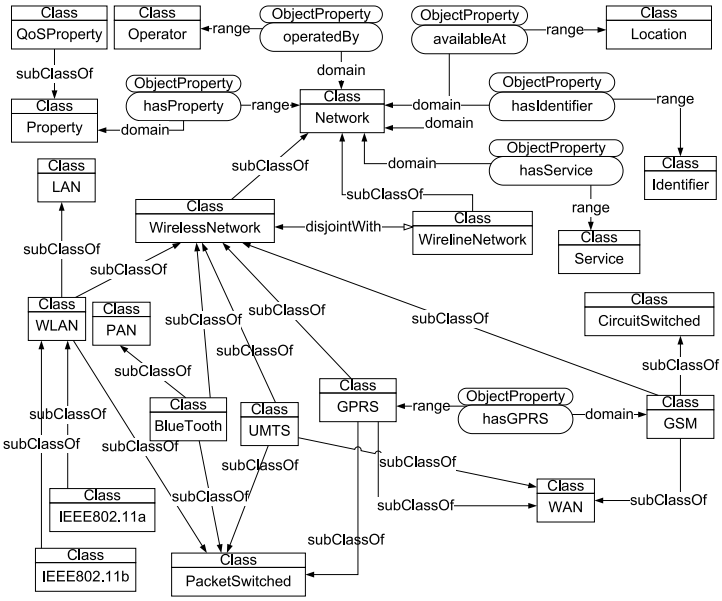
Langattomissa ympäristöissä sovelluskohtaiset ontologiat ovat yhtä tärkeitä kuin missä tahansa ympäristössä, eikä langaton ympäristö sinänsä tuo mitään uutta sovelluskohtaisiin ontologioihin. Toisaalta, jotta ohjelmistoagentit tai muut sovellukset voisivat sopeutua langattomaan ympäristöön ja jakaa tietämystä muiden sovellusten kanssa langattomaan ympäristöön liittyvistä asioista, tarvitaan ontologioita, joiden avulla voidaan kuvata langattomia ympäristöjä. Tällaisia asioita ovat muun muassa langattomat verkot ja niihin liittyvät ominaisuudet. Toinen, vähintäänkin yhtä tärkeä ontologioiden sovelluskohde langattomissa ympäristöissä on päätelaitteiden ominaisuudet. Esimerkiksi FIPA on määritellyt ontologian, jolla voidaan kuvata päätelaitteiden ominaisuuksia [12]. Tässä kuitenkin esittelen vain erään langattomiin verkkoihin ja niiden ominaisuuksien kuvaamiseen tarkoitetun ontologian.

Kuvassa 1 on esitetty osa ontologiasta, jonka avulla voidaan kuvata langattomia verkkoja. Tärkein komponentti ontologiassa on Network-luokka, joka on jaettu subclassOf-relaatiolla langattomiin ja kiinteisiin verkkoihin. Kunkin verkon osalta voidaan lisäksi kuvata sen ominaisuudet (QoS) ja esimerkiksi palvelut, joita ko. verkko tarjoaa. Palvelu voi esimerkiksi olla tekstiviestin lähetyspalvelu, joka toimii GSM-verkossa muttei WLAN-verkossa. Vaikka ontologian peruskomponentit ovat suhteellisen yksinkertaisia, niin todellisten instanssien määrä on laaja. Ontologia esimerkiksi määrittelee käsitteen 'GSMNetwork'. Käytännössä tästä käsitteestä on olemassa satoja ilmentymiä. Esimerkiksi GSM-verkko, jota TeliaSonera tarjoaa Suomessa, on eräs tämän luokan ilmentymä. Tällä ilmentymällä saattaa kuitenkin olla erilaiset ominaisuudet kuin jollakin toisella GSM-verkolla jossain toisessa valtiossa. En tässä mene ontologian yksityiskohtiin. (ks. [19] lisätietoja ontologiasta).

Verkkotyypin ohella erittäin tärkeä asia, joka tulee ottaa huomioon langattomissa ympäristöissä, on muutos. Käytettäessä erilaisia langattomia verkkoja saattaa verkon tarjoama laatutaso vaihdella huomattavasti siirryttäessä verkosta toiseen, mikä on otettava huomioon sovelluksia suunniteltaessa. Laatutaso saattaa vaihdella myös käytettäessä samaa verkkoa johtuen esimerkiksi muuttuvista olosuhteista tai käyttäjän vaihtaessa fyysisesti paikkaa verkon kuuluvuusalueella. Langattomiin ympäristöihin suunniteltujen sovellusten tulee olla suunniteltu siten, että verkon laatutasossa tapahtuviin muutoksiin on varauduttu. Tämä puolestaan tarkoittaa sitä, että verkon laatutaso tulee pystyä kuvaamaan koneen ymmärtämällä tavalla. Verkon tarjoamaa laatutasoa voidaan kuvata QoS ontologialla. FIPA on määritellyt ontologian tätä tarkoitusta varten [14]. Ontologian tärkeimmät käsitteet on esitetty kuvassa 2. FIPA:n määrittelemä ontologia ei kuitenkaan perustu nykyään käytettyihin ontologioiden kuvauskieliin kuten OWL:iin, mutta muunnos FIPA:n määrittelemän ontologian kuvauksen ja OWL kuvauksen välillä on triviaali.

3 Arkkitehtuuri

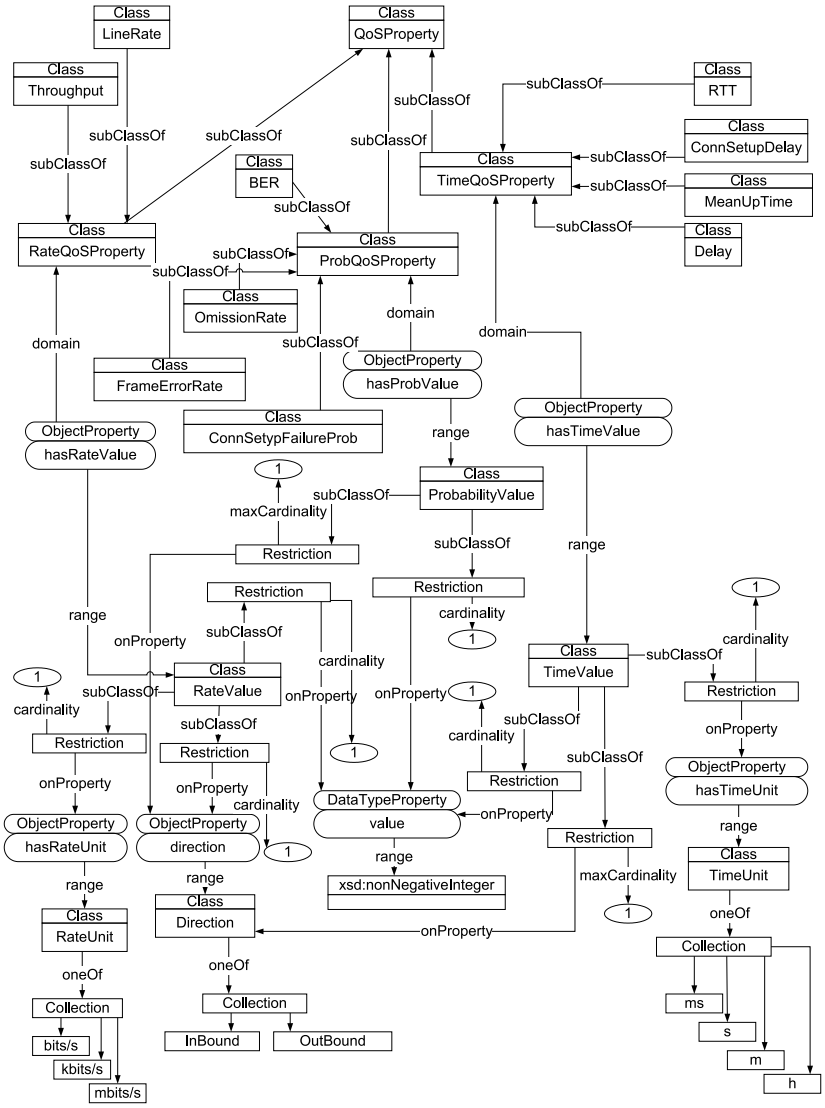
The Foundation for Intelligent Physical Agents (FIPA) on vuonna 1996 perustettu ei-kaupallinen järjestö, jonka tarkoituksena on tuottaa standardeja ohjelmistoagent-



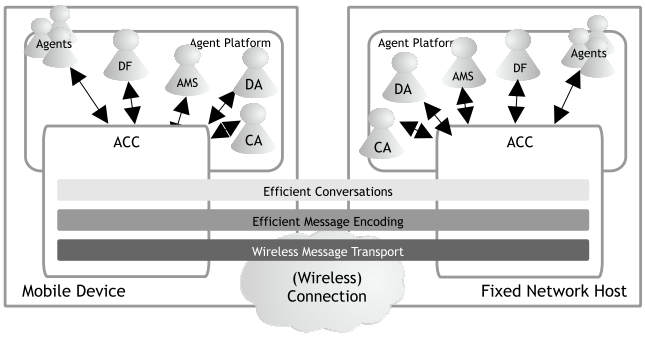
Kuva 1: Langattomien verkkojen kuvaamiseen tarkoitetun ontologian ylimmän tason komponentit

tien väliseen yhteistyöhön, jotta heterogeenisille, eri valmistajien alustoille toteutetut agentit voisivat toimia tehokkaasti yhteistyössä. FIPA:n määrittelemä agenttialusta tarjoaa perustyökalut ohjelmistoagenttien toteuttamiseen. Työ FIPA:ssa keskittyy lähinnä agenttien vuorovaikutukseen, joten FIPA ei ota kantaa siihen, miten agentit toteutetaan sisäisesti. Jotta agenttialusta olisi FIPA-yhteensopiva, sen tulee toteuttaa kolme osakokonaisuutta. Agenttien hallintajärjestelmä (Agent Management System, AMS) on agentti, jonka tehtävä on hallita agenttien suorittamista. Hakemistopalvelu (Directory Facilitator, DF), joka sekin on agentti, nimensä mukaisesti tarjoaa hakemiston, josta voidaan etsiä halutun palvelun toteuttavia agentteja. Agenttien vuorovaikutuskanava (Agent Communication Channel, ACC) tarjoaa palvelun, jonka avulla agentit voivat vaihtaa tietämystä keskenään toteutusriippumattomasti. Tarkemmat kuvaukset näistä komponenteista löytyvät FIPA:n määrittelyistä (ks. [13]).

Edellä luetellut komponentit ovat hyvin yleisiä eivätkä siten ota kantaa siihen, millaisessa ympäristössä agenttien tai niiden avulla toteutettujen sovellusten tulisi toimia. FIPA on laajentanut arkkitehtuuriaan siten, että se soveltuu myös tällaisiin ympäristöihin ja ottaa huomioon langattoman tietoliikenteen erityisvaatimukset. Kuvassa 3 on esitetty viitearkkitehtuuri, jossa yleisten komponenttien lisäksi on esitetty langattoman tietoliikenteen erityisvaatimukset huomioon ottavat komponentit. Kommunikaatioagentti (Communication Agent, CA) toteuttaa edellisessä luvussa kuvatut ontologiat ja niihin liittyviä toiminnallisuuksia. Kommunikaatioagentin toiminnallisuus voidaan jakaa



Kuva 2: Verkon laatutason kuvaamiseen tarkoitetun ontologian ylimmän tason komponentit



Kuva 3: Viitearkkitehtuuri agenttialustalle langattomaan ympäristöön

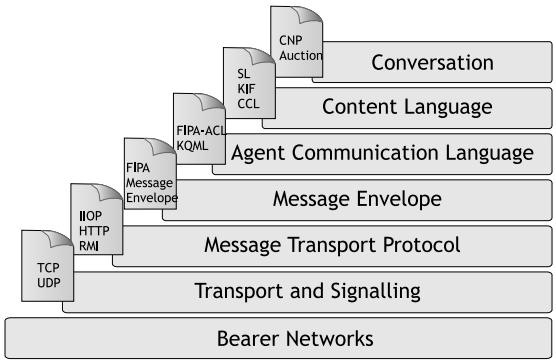
kahteen ryhmään: Ensinnäkin agentin tehtävänä on valvoa tietoliikenneyhteyttä sekä ylläpitää tietoa yhteyden laatutasosta. Kommunikaatioagentti toteuttaa rajapinnan, jonka kautta muut agentit voivat tiedustella esimerkiksi käytössä olevan (langattoman) verkon tarjoamaa laatutasoa käyttäen edellä kuvattua QoS-ontologiaa. Toinen kommunikaatioagentin toiminnallisuus on langattoman yhteyden hallinta. Kommunikaatioagentti voi esimerkiksi avata ja sulkea yhteyden automaattisesti tarpeen mukaan. Lisäksi kommunikaatioagentti tarjoaa rajapinnan, jota käyttäen muut agentit voivat pyytää kommunikaatioagenttia avaamaan tai sulkemaan yhteyden. Varsinaiseen langattomaan yhteyteen liittyvän toiminnallisuuden lisäksi, kommunikaatioagentti tarjoaa toiminnallisuuden, jonka avulla hallitaan erilaisia agenttien vuorovaikutukseen tai sanomien esitystapoihin liittyviä parametreja. Näihin tutustumme tarkemmin seuraavassa luvussa.

Laiteagentin (Device Agent, DA) tehtävänä on hallita päätelaitetta, jossa agenttialusta sijaitsee. Siinä missä kommunikaatioagentti tarjoaa muille agenteille tietoja langattomasta yhteydestä, laiteagentti tarjoaa tietoja päätelaitteesta. Laiteagentti voi myös kontrolloida laitetta. Tällainen toiminnallisuus voi esimerkiksi liittyä laitteen virrankulutuksen säästämiseen.

4 Agenttien välinen vuorovaikutus

Langattomissa ympäristöissä agenttien tulee voida jakaa tietämystään tehokkaasti ja vuorovaikutuksen tulee olla luotettavaa. Siten, vuorovaikutusarkkitehtuuri tulee suunnitella huolellisesti ottaen huomioon langattomien yhteyksien erityispiirteet. Kuvassa 4 on esitetty kerrosmalli, jonka avulla agenttien vuorovaikutusta voidaan mallintaa. Jotta agenttien vuorovaikutus olisi tehokasta ja luotettavaa langattomassa ympäristössä, on mallin jokaisella kerroksella otettava huomioon langattomien yhteyksien erityispiirteet. Tässä luvussa esittelen sekä kerroksien toiminnallisuudet että sen, miten kerrokset voidaan toteuttaa langattomassa ympäristössä siten, että langattoman siirtotien ominaisuudet otetaan huomioon.

Kuljetuskerros (Transport and Signalling) tarjoaa datan siirtopalvelun. Tämän kerroksen tulee olla ja usein onkin läpinäkyvä agenteille, joten en tässä käsittele tätä kerrosta lainkaan. Sano-



Kuva 4: Kerrosmalli agenttien vuorovaikutuksen kuvaamiseen

mien siirtokerros (Message Transport Protocol) määrittelee protokollan sille miten kokonaisia sanomia välitetään agentilta toiselle. Tyypillisesti tämä protokolla määrittelee myös kuljetuskerroksella käytettävän protokollan. Esimerkiksi IIOP-protokolla käyttää TCP:tä kuljetusprotokollana. FIPA on määritellyt HTTP-protokollan ja IIOP-protokollan sanomien siirtoprotokolliksi. Kumpikaan näistä protokollista ei sovellu erityisen hyvin langattomiin ympäristöihin [17, 18]. IIOP-protokollan suorituskyky on parempi kuin HTTP-protokollan, mutta toisaalta HTTP-protokollan toteuttaminen on yksinkertaisempaa kuin IIOP-protokollan. Siten HTTP-protokolla soveltuu paremmin päätelaitteisiin, joissa on vain rajalliset mahdollisuudet toteuttaa vaativia sovelluksia. Suorituskyvyn lisäksi kumpikaan näistä protokollista ei toteuta riittävää luotettavuutta langattomissa ympäristöissä. Siirtoprotokollia, jotka ovat erityisesti suunniteltu agenttien vuorovaikutukseen langattomissa ympäristöissä, ovat muun muassa MAMAv2 [16] ja JICP [5].

Sanoman kirjekuori (Message Envelope) määrittelee siirtoprotokollasta riippumattomalla tavalla erilaisia sanoman reititykseen ja vastaanottajaan liittyviä parametreja. Sanoman kirjekuorella on kaksi pääasiallista tarkoitusta: Ensinnäkin se mahdollistaa sanoman reitityksen ilman, että tähän osallistuvien komponenttien tarvitsee ymmärtää agenttien kommunikointikieltä (Agent Communication Language, ACL). Näin ollen reitityskomponentin ei tarvitse olla agentti, mikä tietyissä olosuhteissa saattaa olla tehokkaampi ratkaisu. Toiseksi, koska sanoman kirjekuori on siirtoprotokollasta riippumaton, voidaan sanoman välittämisessä käyttää erilaisia siirtoprotokollia. Sanoma voidaan esimerkiksi välittää langattoman siirtotien yli käyttäen tähän erityisesti suunniteltua protokollaa. Kiinteässä verkossa puolestaan voidaan käyttää siihen paremmin soveltuvaa protokollaa. Jokaisessa sanomassa tulee olla kirjekuori, joka osaltaan kasvattaa sanoman kokoa, jonka langattomassa ympäristössä tulisi olla mahdollisimman pieni. Toisaalta, kirjekuoren esitystapa tulisi olla mahdollisimman hyvin käytettävään siirtoprotokollaan sopiva. FIPA on ratkaissut tämän ongelman määrittelemällä erilaisia kirjekuoren esitystapoja. Esimerkiksi käytettäessä HTTP-protokollaa voidaan käyttää XML-pohjaista esitystapaa. Langattomiin ympäristöihin FIPA on määritellyt ns. bitti-

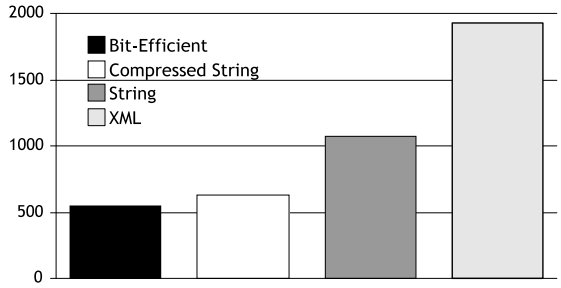
tehokkaan (bit-efficient) esitystavan, jonka tarkoituksena on minimoida kirjekuoren vaatima tilantarve. Tämä esitystapa on hyvin samanlainen kuin FIPA:n määrittelemä esitystapa agenttien kommunikointikielelle.

Ohjelmistoagenttien keskustelujen eteneminen on yleensä täsmällisesti ennaltamääriteltyä. Kun agentti lähettää viestin yhdelle tai useammalle agentille, se liittää tähän viestiin eksplisiittisesti jonkin viestitoimenpiteen. Yhtä viestiä luonnehtii siis aina yksi viestitoimenpide, kuten inform tai agree. Agenttien kommunikointikieli määrittelee sekä viestien semantiikan että syntaksin. FIPA ACL [9] on FIPA:n määrittelemä agenttien kommunikointikieli, joka tarjoaa formaalit määritelmät muun muassa sellaisille viestitoimenpiteille kuten inform, agree ja request. Langattomissa ympäristöissä kielen esitystavan merkitys on suurempi kuin semantiikan, eli viesti on pyrittävä koodaamaan mahdollisimman tehokkaasti. FIPA on määritellyt kolme erilaista syntaksia agenttien kommunikointikielelleen [6, 7, 8]. Tässä keskityn bittitehokkaaseen esitystapaan, joka on suunniteltu erityisesti langattoman tietoliikenteen tarpeisiin.

Bittitehokkaassa ACL-esitystavassa on kaksi tapaa, joilla ACL-viestin vaatimaa tilantarvetta pyritään minimoimaan [6]. Ensinnäkin, sanoma pyritään koodaamaan mahdollisimman pieneen tilaan käyttäen binäärikoodeja merkkijonokoodauksen sijasta. Tyypillisesti tällä koodauksella saadaan sanoman koko noin puoleen merkkijonoihin perustuvaan koodaukseen verrattuna ja noin neljäsosaan verrattuna XML-koodaukseen. Toinen tapa, jolla saavutetaan vieläkin parempi tiivistyssuhde, on älykkään välimuistin käyttö. Tässä menetelmässä sekä sanoman lähettäjä että vastaanottaja tallettavat usein käytettyjä sanomien osia muistiin, ja kun uudessa sanomassa tarvitaan jotain jo käytettyä osaa, voidaan tämä korvata vain viitteellä muistiin. Näin saavutetaan tehokas sanoman esitystapa. Älykkään välimuistin ratkaisun huono puoli on siinä, että se vaatii tiukkaa kytkentää lähettäjän ja vastaanottajan välillä ja lisäksi se vaatii muistia, jota välttämättä ei liikkuvassa päätelaitteessa ole riittävästi. Tosin tässä menetelmässä voidaan muistin käyttö sovittaa päätelaitteen ominaisuuksiin.

Kuvassa 5 on esitetty esimerkkinä erään ACL-sanoman vaatima tilantarve käyttäen erilaisia koodaustapoja. Vaikka perinteisellä tiivistysmenetelmällä päästään lähes samaan tilantarpeeseen kuin käyttäen bittitehokasta koodausta, on bittitehokkaalla koodauksella etuja perinteiseen tiivistysmenetelmään verrattuna. Ensinnäkin, perinteinen tiivistysmenetelmä vaatii melkoisesti laskentatehoa päätelaitteessa, jota ei välttämättä ole käytettävissä. Toiseksi, bittitehokas koodaustapa on suunniteltu siten, että sen käsittely (esim. jäsentäminen) on mahdollisimman tehokasta ja vähän muistia vaativaa. Artikkelissa [18] on analysoitu erilaisia koodaustapoja tarkemmin.

Sisältökielellä (Content Language) määritellään sanoman varsinainen sisältö. Esimerkiksi jos ACL-sanomassa määritellään, että lähettäjäagentti pyytää (request) jotain toista agenttia tekemään jonkin toimenpiteen, niin sisältökielellä esitetään se, mistä toimenpiteestä on kyse. FIPA on määritellyt useita erilaisia sisältökieliä [10], mutta jokaiselle vain yhden esitystavan. Valitettavasti kaikkien FIPA:n määrittelemien sisältökielien esitystapa perustuu joko merkkijonoihin tai XML:ään, joten sisältökielen esittäminen standardilla tavalla kasvattaa sanoman kokoa huomattavasti, vaikka sanoman muut osat olisi koodattu tehokkaasti. Eräs rat-



Kuva 5: ACL sanoman vaatiman tilantarpeen vertailu erilaisilla esitystavoilla

kaisu on käyttää jotakin yleiskäyttöistä tiivistysmenetelmää, joka kyllä koodaa sisältökielen tehokkaasti, mutta edellä esitetyn perusteella tämä ratkaisu ei ole kaikissa tapauksissa mahdollinen. Toinen mahdollinen ratkaisu niissä tapauksissa, joissa esitystapa perustuu XML:ään, on käyttää jotakin binary-XML-esitystapaa [20]. Näillä molemmilla ratkaisuilla on se yhteinen haittapuoli, että koska niitä ei ole standardoitu, ei niiden käyttäminen ole mahdollista avoimessa ympäristössä.

Ohjelmistoagentit keskustelevat usein nojautuen johonkin interaktioprotokollaan [11]. Interaktioprotokolla nimeää järjestetyn joukon viestitoimenpiteitä sekä joitain lisärajoituksia niiden lähettämiselle, kuten kuka aloittaa protokollan tai mihin mennessä johonkin tiettyyn viestiin tulee vastata. Interaktioprotokolla eivät kasvata sanoman kokoa kuten muut vuorovaikutusmallin kerrokset. Interaktioprotokollat on kuitenkin otettava huomioon langattomassa ympäristössä. Keskeinen interaktioprotokollan optimointikeino on vähentää tarvittavia kiertoviiveitä (round-trips). Käytännössä tämä tarkoittaa sitä, että protokolla suunnitellaan siten, että tarvitaan mahdollisimman vähän sellaisia tilanteita, joissa sanoman lähettäjä joutuu odottamaan vastausta sanomaan, ennen kuin voi lähettää uuden sanoman. Toinen mielenkiintoinen optimointikeino erityisesti langattomissa ympäristöissä on protokollan valinta ympäristön perusteella. Esimerkiksi jos langaton yhteys on todella hidas ja viipeet ovat pitkiä, valitaan käytettäväksi yksinkertaista protokollaa, jolla ei välttämättä saavuteta agenttien kannalta optimaalista lopputulosta. Silloin kun langaton yhteys on riittävän nopea, voidaan käyttää saman protokollan monimutkaisempaa versiota, joka esimerkiksi vaatii enemmän sanomia, ja agentit voivat siten päästä parempaan lopputulokseen. Tällainen protokollan kontekstiin sidottu valinta on kuitenkin hyvin sovelluskohtaista, joten väliohjelmiston on vaikea tarjota automaattista tukea tällaiselle toiminnalle.

5 Lopuksi

Tässä artikkelissa käsittelin ohjelmistoagenttiteknologian hyödyntämistä liikkuvassa tietojenkäsittelyssä keskittyen erityisesti vaihteleviin tietoliikenneolosuhteisiin sopeutuvien sovellusten tukemiseen. Ohjelmistoagenttiteknologian katsotaan olevan seuraava kehitysaskel oliopohjaisesta järjestelmien kehittämisestä. Ohjelmistoagenttien avulla voidaan rakentaa itsenäisiä, tavoitehakuisia ja älyk-

käitä hajautettuja järjestelmiä, jotka mahdollistavat uuden tyyppisten, käyttäjäystävällisempien sovellusten toteuttamisen. Yhdistettynä langattomaan tietoliikenneympäristöön, ohjelmistoagenttiteknologia muodostaa hedelmällisen pohjan liikkuvaa käyttäjää tukevien sovellusten toteuttamiseen.

Viitteet

[1] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference. Aug. 2003. W3C Candidate Recommendation, Work in progress.

[2] F. Bellifemine, A. Poggi, and G. Rimassa. JADE — A FIPA-compliant agent framework. In *Proceedings of the 4th International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-99)*, pages 97–108, London, UK, 1999.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[4] D. Brickley and R. V. Guha, editors. *RDF Vocabulary Description Langugae 1.0: RDF Schema*. World Wide Web Consortium, Oct. 2003. W3C Working Draft. Work in progress.

[5] G. Caire, N. Lhuillier, and G. Rimassa. A communication protocol for agents on handheld devices. In *Workshop on Ubiquitous Agents on Embedded, Wearable and Mobile Devices*, Bologna, Italy, July 2002.

[6] Foundation for Intelligent Physical Agents. *FIPA ACL Message Representation in Bit-Efficient Specification*. Geneva, Switzerland, Oct. 2000. Specification number XC00069.

[7] Foundation for Intelligent Physical Agents. *FIPA ACL Message Representation in String Specification*. Geneva, Switzerland, Nov. 2000. Specification number XC00070.

[8] Foundation for Intelligent Physical Agents. *FIPA ACL Message Representation in XML Specification*. Geneva, Switzerland, Oct. 2000. Specification number XC00071.

[9] Foundation for Intelligent Physical Agents. *FIPA Communicative Act Library Specification*. Geneva, Switzerland, Nov. 2000. Specification number XC00037.

[10] Foundation for Intelligent Physical Agents. *FIPA Content Languages Specification*. Geneva, Switzerland, Oct. 2000. Specification number XC00007.

[11] Foundation for Intelligent Physical Agents. *FIPA Interaction Protocol Library Specification*. Geneva, Switzerland, Oct. 2000. Specification number XC00025.

[12] Foundation for Intelligent Physical Agents. *FIPA Device Ontology Specification*. Geneva, Switzerland, Apr. 2001. Specification number XC00091.

[13] Foundation for Intelligent Physical Agents. FIPA Home Page, 2001. Available from World Wide Web: <`http://www.fipa.org`>.

[14] Foundation for Intelligent Physical Agents. *FIPA Quality of Service Specification*. Geneva, Switzerland, Aug. 2003. Specification number XC00094.

[15] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Padova, Italy, 1993. Kluwer Academic Publishers.

[16] H. Helin. Supporting nomadic agent-based applications in FIPA agent architecture. PhLic. Thesis, Series of Publications C, Number C-2001-63, University of Helsinki, Department of Computer Science, Helsinki, Finland, Dec. 2001.

[17] H. Helin. *Supporting Nomadic Agent-based Applications in the FIPA Agent Architecture*. PhD thesis, University of Helsinki, Department of Computer Science, Helsinki, Finland, Jan. 2003.

[18] H. Helin and M. Laukkanen. Performance analysis of software agent communication in slow wireless networks. In R. Luijten, E. Wong, K. Makki, and E. K. Park, editors, *Proceedings of the Eleventh International Conference on Computer Communications and Networks (ICCCN'02)*, pages 354–361. IEEE, Oct. 2002.

[19] H. Helin and M. Laukkanen. Wireless network ontology. In *Proceedings of the Wireless World Research Forum Ninth Meeting (WWRF#9)*, Zürich, Switzerland, July 2003.

[20] Wireless Application Protocol Forum. *Binary XML Content Format Specification*, Nov. 1999. Version 04-Nov-1999.