Ontologiat ja agenttien yhteistoimintakielet

Teppo Pennanen

Status: draft
Security: public

Contents

1 Johdanto . . . . . .--

2 Agentit . . . . . .--

2.1 Agenttiparadigma . . . . . .--

2.2 Agentti . . . . . .--

2.3 Agenttijärjestelmä . . . . . .--

3 Agenttikielet . . . . . .--

3.1 Agenttikieli . . . . . .--

3.2 KQML . . . . . .--

3.3 FIPA ACL . . . . . .--

4 Agenttien Käyttöesimerkki: RoboCup . . . . . .--

4.1 Yleistä . . . . . .--

4.2 Simulaattoriliiga . . . . . .--

5 Yhteenveto . . . . . .--

1 Johdanto

Agentit tarjoavat uuden tavan tutkia ja rakentaa hajautettuja järjestelmiä. Agenttia voidaan pitää oliona, jolla on kolme ominaisuutta: autonomisuus, joustavuus ja vuorovaikutus. Agenttialustat tarjoavat agenteille rajapinnan, hallintakeinot sekä palvelujen hallintaa. Agentit kommunikoivat Agenttikielillä, jotka ovat korkean tason kieliä. Useat agentit ja agenttialustat muodostavat Moniagenttijärjestelmiä, joissa agentit voivat toimia ja vaikuttaa sekä joiden välillä agentit voivat liikkua. Johtuen agenttien ominaisuuksista, ne soveltuvat hyvin hajautettuihin heterogeenisiin järjestelmiin.

Agenttiteknologian termejä on vaikea määritellä, koska yleisesti ei haluta sulkea asioita pois määrittelemällä liian tarkkaan. Tutkimus ja standardointi on aiheesta vielä kesken.

Toisessa kappaleessa tarkastellaan Agentteja, agenttialustoja ja moniagenttijärjestelmiä tarkemmin. Kappale kolme sisältää asiaa agenttikielestä ja siinä on esimerkkinä esitelty agenttikielet KQML ja FIPA ACL. Kappaleessa neljä esitellään käytännön sovellus agenteista: RoboCup. Kappaleessa viisi on yhteenveto tästä tutkielmasta.

2 Agentit

2.1 Agenttiparadigma

Agenttiteknologia ei ole täysin uusi ja erilainen teknologia-ala vaan enemmänkin useamman teknologian yhteistuotos. Agenttien toteutustapaa ei ole sidottu ja niissä voidaan käyttää esimerkiksi olio-, logiikka— ja hajautettuja paradigmoja tai useampia eri paradigmoja.

Agenttiteknologian termit ovat vaikeita määritellä koska ihmisillä on erilaisia näkemyksiä aiheesta. On helppoa verrata agentteja olioihin ja niillä onkin paljon yhteista. Agenteilla on kuitenkin myös paljon erilaisuuksia olioiden kanssa.

Agentit ja agenttialustat eivät sulje pois hajautettuja arkkitehtuureja kuten CORBA tai Java RMI. Agentit toimivat näiden yläpuolella ja agenttialustassa on implementoitu tuki hajautetulle järjestelmälle. Alla olevasta kuvasta käy ilmi agenttien ja agenttialustojen sijainti hajautetussa järjestelmässä /2/.

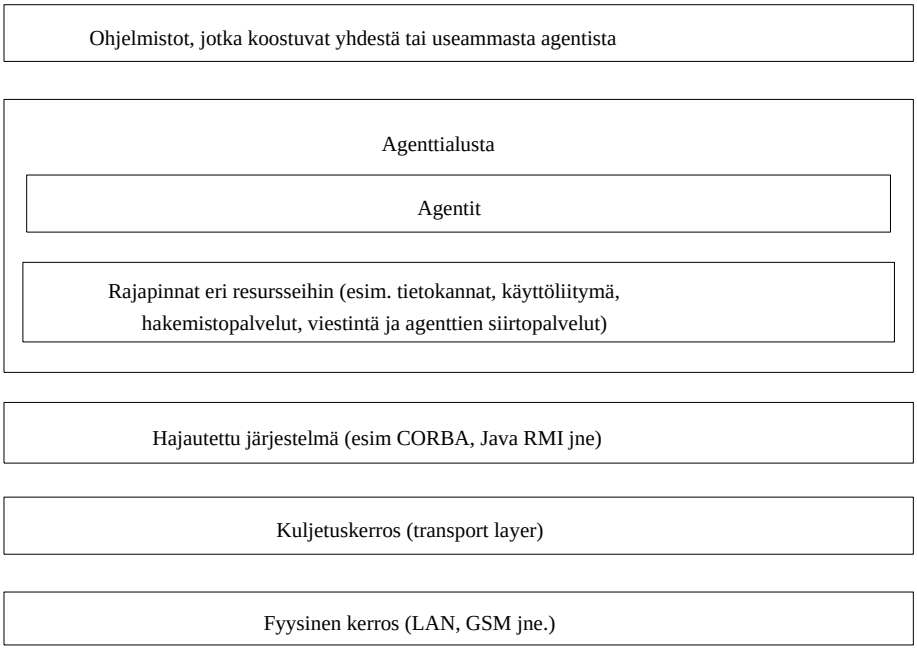

Figure 1. Yleiskuva agenttien ja agenttialustan muodostamasta kokonaisuudesta.

2.2 Agentti

Agenttien ominaisuuksia

Agentti voi koostua erilaisista osajoukoista seuraavanlaisista ominaisuuksista

- Autonominen (autonomous)
- Vuorovaikutuskykyinen (interactive)

- Joustava (adaptive/reactive)
- Liikkuva (mobile)
- Puolestatoimija (proxy)
- Järkeilevä (rational)
- Maalihakuinen (proactive)
- Älykäs (intelligent)
- Pysyvä (temporally continuous)
- Luonteen omaava (character)
- Oppimiskykyinen (able to read)
- Yhteistyökykyinen (co-operative)
- Vikasietoinen (rugged)
- Luotettava (trustworthy)

Yksittäisellä agentilla ei tarvitse olla kaikkia edellä mainittuja ominaisuuksia, mutta yleisesti kolmea ensimmäistä pidetään tärkeimpinä. Koska ominaisuusjoukko on aika laaja, on agentin tarkan määrittelyn antaminen hankalaa /1/.

Agentit ja oliot

Agentit ja oliot muistuttavat toisiaan monella tapaa. Molemmilla on identiteetti, tila, käyttäytyminen ja rajapintoja. Johtuen näistä yhtäläisyyksistä agenteilla ja olioilla on samankaltaisia tarpeita liittyyen erilaisten vaatimusten määrittelyyn. Jotta määrittelyssä ei tarvitsisi tehdä samaa työtä useaan kertaan kiinnostaa agenttien ja olioiden suhteet tiedeyhteisöä

Agentit ovat olioita kehittyneempiä (entiteettejä) ja niillä on yleensä enemmän (älykästä) toiminnallisuutta kuin oliolla. Agentit ovat itsenäisiä ja reagoivat ympäristönsä kanssa, niillä on omat motiivit, tavoitteet, rajoitteet sekä ne pystyvät mukautumaan ja oppimaan.

Vaikka agentteja voidaan jakaa eri luokkiin, agentit mielletään yleensä agentteiksi eikä tietyn tyyppisiksi agenteiksi. Olioiden on aina oltava tietyn tyyppisiä. Agentti voi toki erikoistua ja muistuttaa jonkin tyyppistä olioita. Lisäksi olioille tyypillinen perintä ei kuulu agenteille vaikkakaan sitä ei ole kielletty.

Agentit käyttävät ilmaisuvoimaisia agenttikieliä (Agent Communication Language). Agenttikielet jaetaan kahteen ryhmään KQML ja FIPA ACL. Ne ovat osittain samankaltaisia

mutta eroavat yksityiskohdiltaan. Agenttikielistä kerrotaan enemmän kappaleessa kolme.

Agentteja voidaan luokitella rooliensa tai käyttäytymisensä mukaan :

- Oppiva agentti (learning agent) keskittyy mallien tunnistamiseen sekä oppimisen voimistamiseen.

- Tarkoitusperäinen agentti (intentional agent) tekee suunnitelmia ja toteuttaa ne jonkin (oman) tavoitteen saavuttamiseksi. Se käyttää puhe-toiminta pohjaista kommunikaatiota (speech-act-based communication) ja heijastavaa järkeilyä (reflective reasoning).

- Sosiaalinen agentti (social agent) on erikoistunut yhteistoimintaan toisten agenttien (tai vaikka olioiden) kanssa tiimeissä tai ryhmissä. Sosiaaliseksi agentiksi voidaan sanoa myös agenttia, jolla on persoonallisuus.

- Liikkuva agentti (mobile agent) on agentti, joka pystyy liikkumaan ympäristöstä toiseen esimerkiksi liikkuvan käyttäjän mukana.

- Tietoagentti (information agent) kerää tietoa yhdestä tai useammista mahdollisesti heterogeenisestä lähteestä ja tarjoaa sitä edelleen eteenpäin käytettäväksi.

- Käyttöliittymäagentti (interface agent) osaa kommunikoida ihmisten kanssa.

- Reaktiivinen agentti (Reactive Agent) reagoi ympäristön muutoksiin tapahtumapohjaisesti.

- Välittäjäagentti (Middle Agent) toimii middlewarena ja voi olla esimerkiksi välittäjä (broker) toisten agenttien etsiessä ajonaikaisesti palvelua.

- Tulkkiagentti (Translator Agent) toimii tulkkina kahden olin välissä tulkkaamassa tietoa toiselta toiselle.

- Reititysagentti (Router Agent) on mukana välittämässä tietoa paikasta toiseen.

Tärkeimpiä agenttityyppejä ovat käyttöliittymäagentti, reaktiivinen agentti, liikkuva agentti, oppiva agentti ja välittäjäagentti /1/.

2.3 Agenttijärjestelmä

Agenttialusta

Agenttijärjestelmät koostuvat yhdestä tai useammasta agenttialustasta. Agenttilalusta (agent platform) on kokoelma läheisesti toisiinsa liittyviä palveluja. Se mahdollistaa agenttien toiminnan tarjoamalla rajapinnat alaspäin laitteistoon ja verkkoon. Agenttien tulee rekisteröityä pystyäkseen olemaan vuorovaikutuksessa alustan muiden agenttien kanssa ja muiden agenttialustojen kanssa.

Agenttialusta koostuu kolmesta osasta: agenttien hallintajärjestelmästä (AMS, Agent Management System), agenttien kommunikaatiokanavista (ACC, Agent Communication Channel) ja hakemiston hallitsijasta (Directory Facilitator) /4/.

Agenttien hallintajärjestelmä (AMS) on agentti, joka hoitaa agenttien luomisen, tuhoamisen, keskeyttämisen (suspension), vapautuksen (redemption), todentamisen ja vaeltamisen. AMS-agentteja saa olla vain yksi ja niitä pitää olla yksi kussakin agenttialustassa. AMS pitää yllä listaa agenteista, jotka ovat sen hallitsemalla agenttialustalla sekä agenttialustoista joiden kanssa se on tekemisissä. Koska eri agenttialustoilla voi olla eri ominaisuuksia, AMS:ltä voidaan kysyä agenttialustan profiilia. AMS on ylin auktoriteetti agenttialustalla ja se voi pyytää agenttia lopettamaan (quit) toimintansa alustalla ja jos pyyntöä ei totella, se voi tuhota agentin.

Agenttien kommunikaatiokanava (ACC) on agentti joka, agenttien hallintajärjestelmää hyväksi käyttäen, välittää tietoa agenttien väleillä agenttialustan sisällä ja muiden agenttialustojen kanssa. Jokaisella agentilla on pääsy ainakin yhteen ACC:iin ja vain agenteille osoitetut sanomat pääsevät AAC:lle. Kun agentti haluaa kommunikoida toisella agenttialustalla olevan agentin kanssa se voi joko pyytää paikallista AAC:ia välittämään sanoman kohdeagentille tai se voi ottaa yhteyttä suoraan kohdeagentin agenttialustan ACC:iin ja pyytää sieltä sanoman perille viemistä.

Hakemiston hallitsija (DF) on agentti, joka tarjoaa puhelinluettelon keltaisten sivujen kaltaista palvelua agenteille. Se pitää yllä tietoa agenteista ja niiden tarjoamista palveluista. DF tarjoaa palvelut rekirteröitymiselle, etsimiselle, rekisteröinnin poistamiselle sekä muutoksille. Sen luotetaan pitävän tarkkaa täydellistä ja ajantasaista listaa agenteista. Agenttialustalla tulee olla ainakin yksi DF (oletus DF), mutta siellä voi olla useampiakin. DF voi rajoittaa agenttien pääsyä sen tietoihin ja se varmistaa ainakin kaikkien niiden agenttien luvat, jotka ilmoittavat agenttien tilojen vaihdosta. DF:t voivat rekisteröityä toisilleen ja AMS sekä ACC voivat rekisteröityä DF:lle.

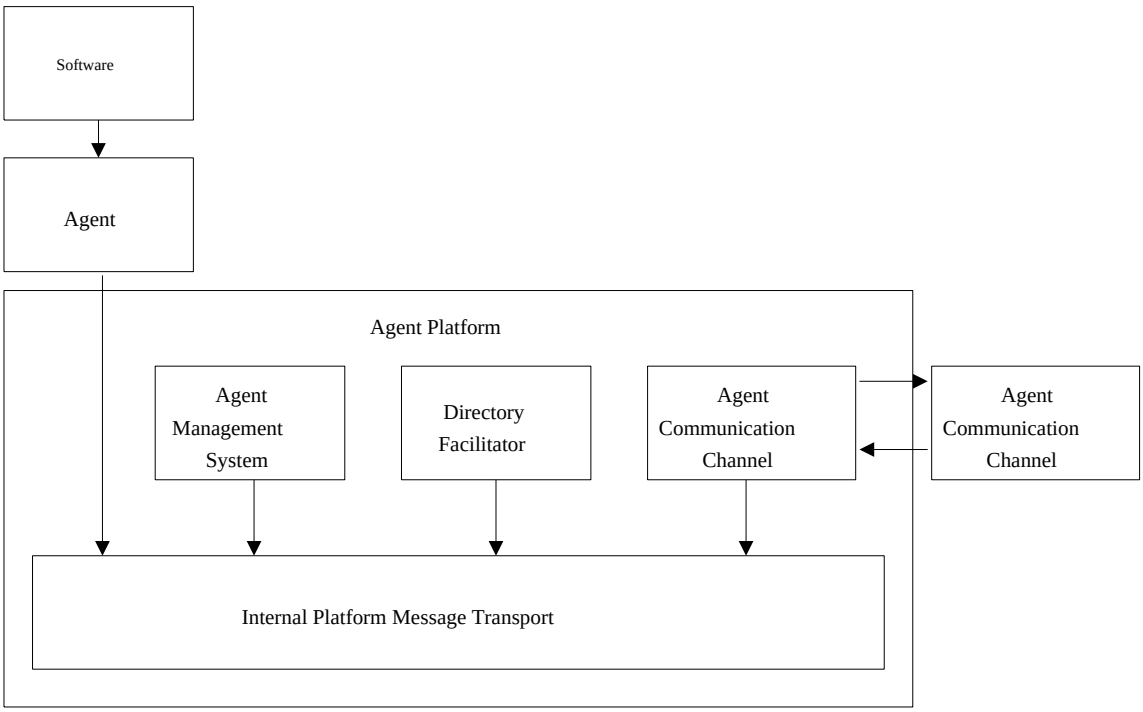

Figure 2. Agenttialusta

Alla on kuva useamman agenttialustan muodostamasta moniagenttijärjestelmästä. Siinä on kaksi agenttialustaa, joista oikeanpuoleisella on kaksi DF:ia. Agentti B on rekisteröitynyt kahdelle eri DF:lle joista toinen on eri agenttialustalla. Agentti D on rekisteröitynyt molemmille oikeanpuoleisen agenttialustan DF:lle /3/.

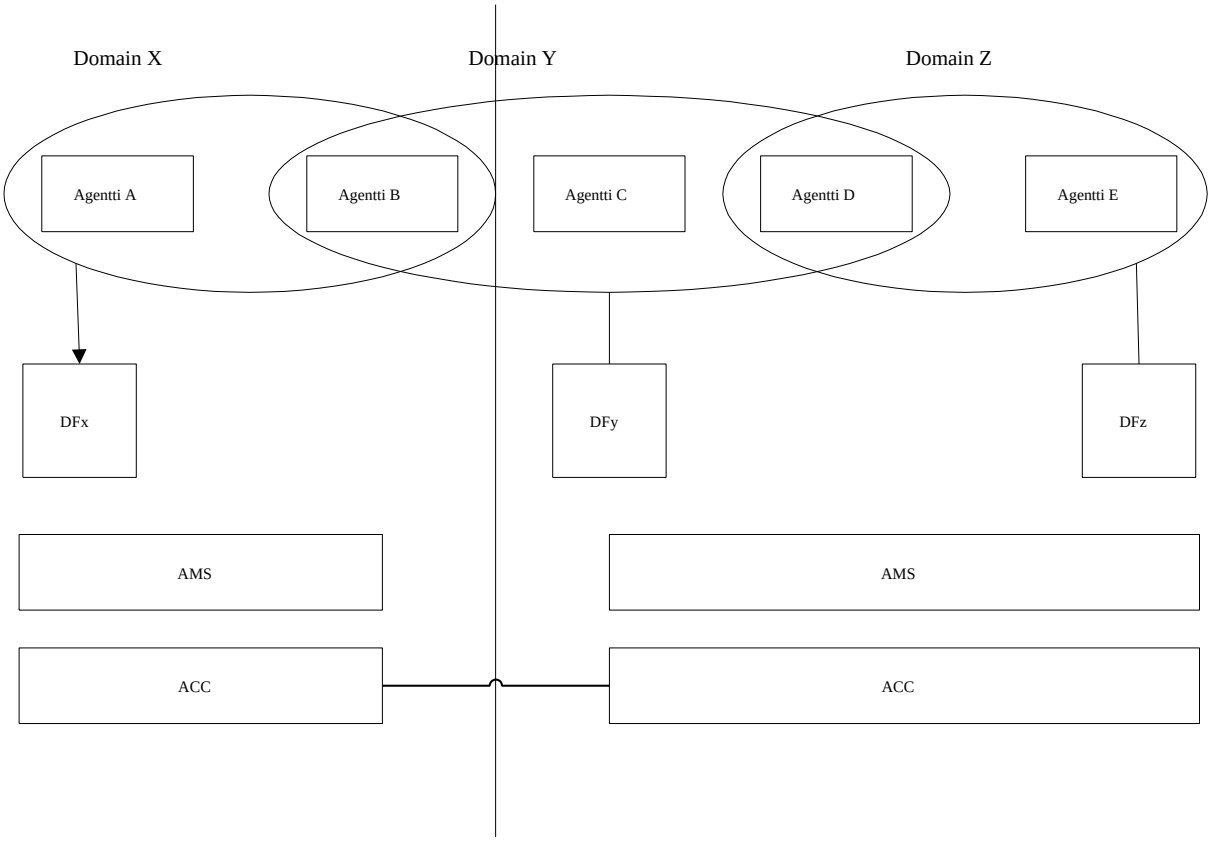

Figure 3. Moniagenttijärjestelmä

Agenttien elinkaari

Agentit ovat fyysisesti olemassa agenttialustalla ja käyttävät agenttialustan palveluja. Koska jokainen agentti on prosessi, sillä on elinkaari. FIPA määrittelee agentin elinkaarimallille seuraavia oletuksia:

- Agentti on aina sidottu agenttialustaan, samoin sen elinkaari on sidottu yhteen agenttialustaan.
- Agentin elinkaarimalli on riippumaton sovelluksesta. Malli määrittelee vain agentin tilat ja tilamuutokset.
- Agenttimalli on ilmentymäorientoitunut eli se käsittää agentin omaavan ainutlaatuisen tunnuksen ja se käsittää agentin olevan itsenäisesti suorituksessa.
- Jokainen agentti on vain yhdessä tilassa yhdellä hetkellä ja vain yhdessä agenttialustassa.

Elinkaarimallissa on kuvattuna seuraavat agentin toiminnot: luominen (create), herättäminen (invoke), tuhoaminen (destroy), hallittu lopettaminen (quit), keskeyttäminen (suspend), jatkaminen (resume), odotuttaminen (wait), herättäminen (wake-up). Lisäksi mallissa on liikkuville agenteille määritelty liikkuminen (move) ja suoritukseen palaaminen (execute). Agentin elinkaaren tilat ja siirtymät on kuvattu alla olevassa kuvassa. Siirtumissä on suluissa mainittu, ketkä voivat määrätä siirron /4/.

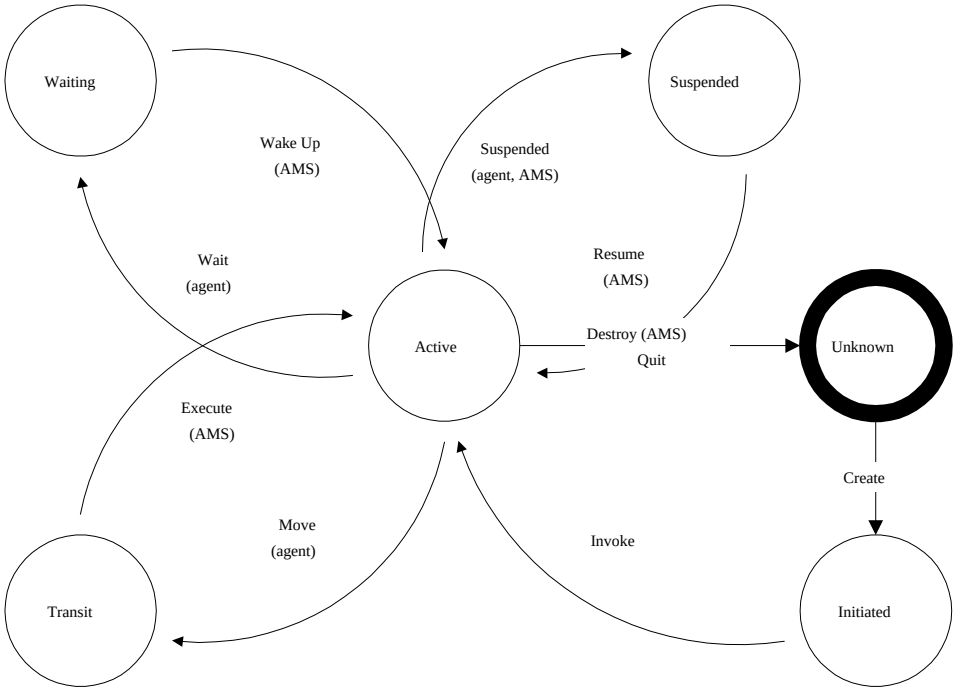

Figure 4. Agentin elinkaaren tilat ja siirtymät

Agenttien rekisteröityminen

Agentin kotiagenttialustaksi (HAP, Home Agent Platform) nimitetään sitä agenttialustaa, jossa agentti on luotu. HAP on vastuussa luodun agentin henkilöllisyyden takaamisesta. Jokaisella agentilla tulee olla yksi HAP ja agentti on itse vastuussa siitä, että HAP:in AMS:lla on agentin voimassa oleva osoite. Agentin täytyy olla aina rekisteröityneenä HAP:iinsa.

Jotta agentti voi tulla rekisteröidyksi AMS:lle tarvitaan agentin globaali tunniste (GUID, Globally Unique Agent Identifier) ja sen paikallinen osoite. Agentti voi tulla rekisteröidyksi AMS:lle kolmella tavalla /4/:

- Agentti on luotu agenttialustalle.
- Agentti on muuttanut agenttialustalle.
- Agentti on erikseen rekisteröitynyt agenttialustalle. Tämä vaatii sen, että agenttialusta tukee dynaamista rekisteröintiä ja on halukas rekisteröimään agentin.

3 Agenttikielet

3.1 Agenttikieli

Agenttien asenteet

Agenttien kommunikaation sisältö on merkityksellisiä lauseita agentin tiedoista ja ympäristöstä. Jotta keskustelulle pystytään antamaan merkitystä, täytyy olettaa, että agentilla on joku seuraavista asenteista lausejoukkoa kohden /3/:

- Usko (Belief). Lauseiden, jotka agentti hyväksyy (olivatpa ne tosia tai ei), se uskoo olevan (tällä) hetkellä totta. Väitteet, jotka agentti uskoo olevan epätosia esitetään uskomalla väitteen negaation olevan totta.

- Epävarmuus (Uncertainty). Ehdotukset, jotka agentti hyväksyy, eikä tiedä lauseiden olevan tosia tai epätosia. Lauseiden uskotaan olevan enemmän tosia kuin epätosia. Lauseet, joiden uskotaan olevan enemmän epätosia kuin tosia pidetään epävarmasti ehdotuksena olevien lauseiden negaationa. Epävarmuus sallii agenttien tulevan mukaan, vaikka ei olisikaan täysin samaa mieltä.

- Aikomus (Intention). Ehdotukset, jotka agentti hyväksyy tällä asenteella, agentti toivoo olevan tosia ja niiden ei tällä hetkellä yleisesti uskota olevan tosia. Tällä asenteella agentti yrittää muuttaa maailmaa halujensa mukaan.

Speech-Act

Agenttien kommunikointi perustuu puhe-toiminta -teorialle (Speech-Act theory). Teoria on ihmispuheen kielellisen analyysin tulos. Ideana on, että puhuja ei ainoastaan tee lausuntoja vaan suorittaa myös toimintoja lauseilla. Agentti pyrkii vaikuttamaan kommunikoinnilla toisiin agentteihin ja saamaan ne tekemään tehtäviä jotka edesauttavat pyytävän agentin omia haluja. Hajautetussa järjestelmässä tämä on yleensä ainoa tapa agentin saada mitään aikaan. Toimintoja joita agentti suorittaa näin kommunikoimalla kutsutaan viestintätoimenpiteeksi (CA, Communicative Act) ja se on erityisjoukko agenttien toimenpiteissä (Actions).

Sanoman lähetys lähtee siitä, että agentilla on jokin tavoite (goal). Saavuttakseen tavoitteen agentti tekee päätöksen (Intent) toiminnasta. Jos päätös on sellainen, jota se ei yksin pysty toteuttamaan, agentti tekee joukon viestitoimintoja, joiden avulla se pystyy pääsemään tavoitteeseen, muodostaa niistä sanomia ja lähettää ne oikeille tahoille /3/.

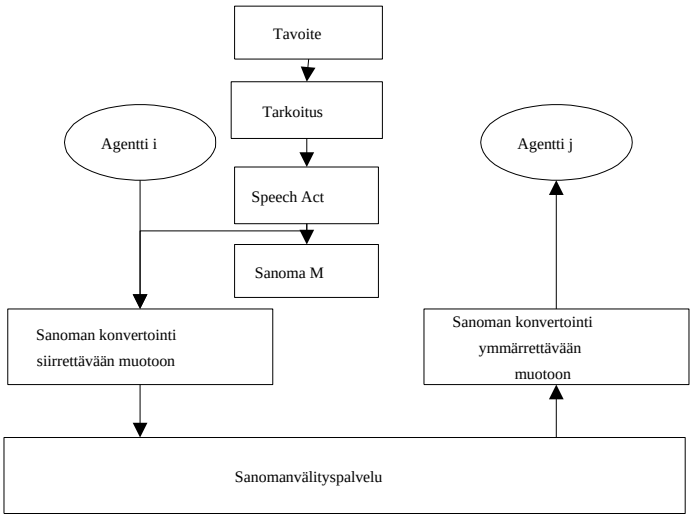

Figure 5. Sanoman muodostus, lähetys ja vastaanotto

Agenttikieli

Agenttikieli (ACL, Agent Communication Language) on korkean tason kieli joka koostuu kolmesta osasta: vuorovaikutusprotokollasta (conversational protocol), sisältökielestä (ADL, architecture description language) ja ontologiasta (Ontology, nature of existence). Agenttikieli on olioiden sanomienlähetystä korkeammalla tasolla oleva kieli jossa käytetään puhetoimintoa (speech act). Agentit voivat esimekiksi kysyä palveja (ASK), kertoa palveluista (TELL) ja tilata palveluita (SUBSCRIBE).

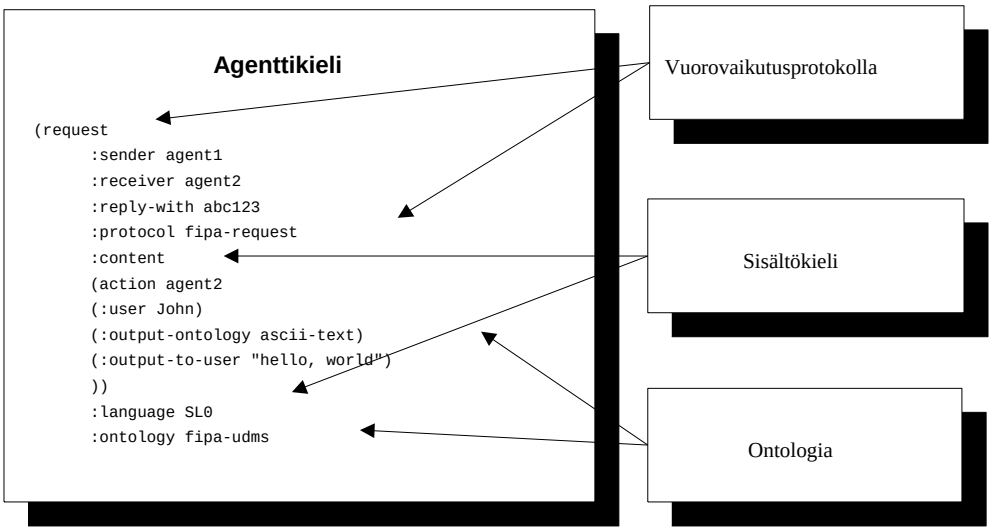

Figure 6. Esimerkki agenttikielestä (KQML ja FIPA ACL)

Vuorovaikutusprotokolla määrittelee oikean (laillisen) sanomien järjestyksen. Se määrittää tapahtumaketjun, jolla on selkeät alku- ja loppupisteet. Protokollaa seurataan määrittelyjen mukaan ja se päättyy hyväksyttyyn tai hylättyyn lopputilaan, riippuen vasteista ja käytetyistä protokollista.

Sisältökielten avulla agentit lähettävät toisilleen erilaisia väittämiä ja loogisia lauseita. Kielessä

on määritelty tarkkaan käytettävissä olevat notaatiot. Semanttinen merkitys lauseille tulee ontologian kautta /2/, /3/.

Ontologia on oppi olevaisesta. Agenttien yhteydessa ontologia tarkoittaa kuitenkin tiedon jakamista, erilaisten käsitysten kokoelmaa. Ontologian avulla yritetään määritellä yhteisiä asioita ja siten saada selville lauseen semanttinen sisältö.

Esimerkiksi jos agentti 1 pyytää agentilta 2 kuvaa sitrushedelmästä. Agentti 2 saa pyynnön, mutta ei löydä yhtään kuvaa sitrushedelmästä. Agentti 2 tarkistaa ontologia-agentilta (käytetyst ontologiasta) sitrushedelmän alalajeja ja saa vastaukseksi mm. appelsiinin ja sitruunan. Se löytää kuvan appelsiinista ja lähettää sen vastauksena /4/.

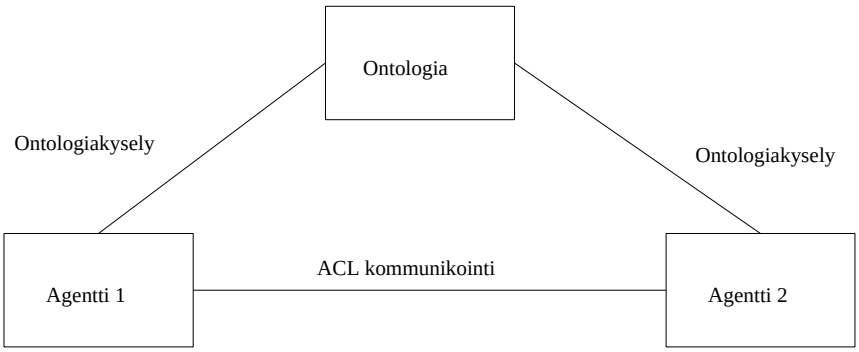

Figure 7. Agenttien kommunikointi ja ontologia

3.2 KQML

External Interfaces Working Group on vuonna 1990 perustettu joukko tutkijoita jotka tutkivat hajautettuja järjestelmiä ja tekoälyä. Ryhmä perustettiin osana DARPA Knowledge Sharing Effort järjestöä kehittämään protokollia itsenäisten tietojärjestelmien tiedonvaihtoon. Päätuloksena syntyi KQML (Knowledge Query and Manipulation Language).

KQML on yleisesti käytetty agenttikieli. Se määrittelee agenttien välisen kommunikaation syntaksin sekä joukon toimintoja, joita kaikkien KQML:ää tukevien agenttien tulisi ymmärtää.

Sanoma

KQML-sanomassa on seitsemän parametria. Ne ovat listattuna ja selitetty seuraavassa taulukossa.

Parametri	Tarkoitus
:sender	Parametri sis ältää sanoman l ähett äneen agentin tunniste.
:receiver	Parametri sis ältää vastaanottajan tunnisteen.
:content	Parametri kertoo sanoman sis ällön tai toiminnan kohteen eli viestinn ällisen toiminnan.

:reply-with	Parametrin ja :in-reply-to -parametrin avulla pystyt ään kytkem ään pyynn öt ja vastaukset toisiinsa.
:in-reply-to	Parametin ja :reply-with -parametrin avulla pystyt ään kytkem ään pyynn öt ja vastaukset toisiinsa.
:language	Parametri m äärittelee parametrin :content kielen.
:force	Parametrilla l ähett äjä ilmoittaa tuleeko se koskaan kielt ämä än toiminnon merkityst ä. Parametri voi saada arvot `permanent` tai `tentative`.

Table 1. KQML:n viestinnälliset toiminnat

Parametri `:content` määrää toiminnan ja se voi saada 31 eri arvoa. Nämä arvot ovat: achieve, advertise, ask-about, ask-all, ask-if, ask-one, break, broadcasta, broker-all, broker-one, deny, delete, delete-all, delete-one, discard, eos, error, evaluate, forward, generator, insert, monitor, next, pipe, ready, recommend-all, recommend-one, recruit-all, recruit-one, register, reply, rest, sorry, standby, stream-about, stream-all, subscribe, tell, transport-address, unregister, untell /5/.

Protokollat ja sisältökieli

KQML ei määrittele pakollisia protokollia vaan protokollat tulisi optimoida järjestelmälle sopiviksi.

KQML:ssä voidaan sopia sisältökieli joka voi olla esimerkiksi SL (Semantic Language), KIF tai Prolog.

Agenttien toiminta

KQML:n yhteydessä on määritelty Agentille esimerkinomaisesti erilaisia politiikkoja, joita niiden tulisi noudattaa

- rehellisyys (honesty), sanoman semantiikan tulisi olla sovellettavissa lähettäjään.
- hyväuskoisuus (gullibility), agentit omaksuvat toisten agenttien väittämät, jos ne käyvät yksiin omien kanssa.
- vastaanottavuus (responsiveness), agentti vastaa kaikkiin saamiinsa sanomiin.
- empatia (empathy), agentti pystyy itse päättelemään, mitä toimintoja muut haluavat, ilman explisiittisiä ohjeita.
- asiallisuus (pertinence), agenttti ei lähetä sanomia, joiden se ei luule auttavan muita.
- Identiteetti (identity), agentit eivät koskaan rekisteröidy jonkun toisen nimellä.

3.3 FIPA ACL

FIPA:n (Foundation for Intelligent Physical Agents) on ei kaupallinen järjestö, jonka tehtävänä on edistää agentteihin perustuvia sovelluksia, palveluja ja laitteita. FIPA pyrkii tavoitteeseensa tuottamalla yleisesti hyväksyttyjä spesifikaatioita joiden pääpaino on agentti-pohjaisten sovellusten keskinäisen toiminnan maksimoimisessa. FIPA on tehnyt oman versionsa agenttikielestä: FIPA ACL.

Sanoma

FIPA ACL sanoma koostuu yhdestätoista parametrista jotka on kuvattu alla olevassa taulukossa:

Parametri	Tarkoitus
:sender	Communicative actin l ähett äneen agentin tunniste.
:receiver	Parametri sis ältää vastaanottajan tai vastaanottajien tunnisteet.
:content	Parametri kertoo sanoman sis ällön tai toiminnan kohteen eli viestinn ällisen toiminnan.
:reply-with	Parametrin ja :in-reply-to -parametrin avulla pystyt ään kytkem ään pyynn öt ja vastaukset toisiinsa.
:in-reply-to	Parametin ja :reply-with -parametrin avulla pystyt ään kytkem ään pyynn öt ja vastaukset toisiinsa.
:envelope	parametri sis ältää viitteit ä sanomaan liityv ään hy ödylliseen tietoon, kuten esimerkiksi l ähetysaika tai —reitti.
:language	Parametri m äärittelee parametrin :content kielen.
:ontology	parametri m äärittelee sanoman ontologian
:reply-by	parametri kertoo ajan hetken, johon mennessa kyselij ä haluaisi vastauksen.
:protocol	parametri m äärittelee mit ä protokollaa noudatetaan.
:conversation-id	parametrin avulla tunnistetaan, mihin keskusteluun sanoma kuuluu.

Table 2. FIPA:n määrittelemät viestinnälliset toiminnat

Parametri `:content` voi FIPA:n standardin mukaan saada 20 eri arvoa. Arvot ovat: accept-proposal, agree, cancel, cfp, confirm, disconfirm, failure, inform, inform-if, inform-ref, not understood, propose, query-if, query-ref, refuse, reject-proposal, request, request-when, request-whenever ja subscribe /3/.

Protokollat

FIPA määrittelee standardissaan seitsemän protokollaa. Nämä protokollat ovat request, query, request-when, contract-net, iterated-contract-net, auction-english ja auction-dutch protokollat. Seuraavassa kuvassa on esimerkinomaisesti kuvattuna request protokolla /3/.

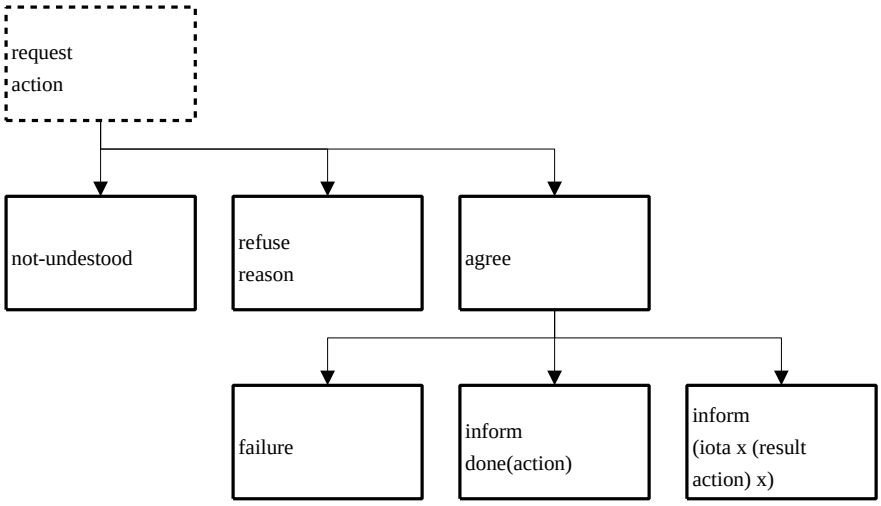

Figure 8. Esimerkkinä FIPA:n määrittelemä request-protokolla

Yhteensopivien agenttien minimivaatimukset

FIPA määrittelee seuraavanlaiset minimivaatimukset agenteille, jotta niitä voidaan sanoa FIPA ACL —yhteensopiviksi /3/.

- Agentin tulee lähettää *not-understood*, jos se ei ymmärrä sanomaa tai pysty vastaamaan siihen. Agenttien tulee myös valmistua vastaanottamaan käsittelemään *not-understood*.

- Agentti voi toteuttaa minkä tahansa osajoukon esimääritellyistä sanomista ja protokollista. Toteutuksen tulee olla kyseisen toiminnan semanttisen määrittelyn mukainen.

- Agentin, joka käyttää FIPA:n määrittelemiä viestinnällisiä toimintoja, tulee toteuttaa ne kuvausten mukaisesti.

- Agentin tulee pystyä rakentamaan sanoma siten, että se on syntaktisesti oikein ja vastaa semanttisesti agentin omaa tarvetta. Vastaavasti agentin tulee pystyä purkamaan sanoma oikein.

- Agentin tulee käyttää standardin mukaista ACL.ää ja ontologiaa

- Agentin ei tarvitse toteuttaa yhtään standardiprotokollista eikä sitä rajoiteta käyttämästä muita protokollanimiä. Jos agentti kuitenkin käyttää standardiprotokollanimiä, tulee protokollat toteuttaa määritelmien mukaan.

Sisältökielet

FIPA määrittelee SL:n (Semantic Language) olemaan sisältökieli /3/.

4 Agenttien Käyttöesimerkki: RoboCup

4.1 Yleistä

Vuonna 1992 professori Alan Mackworth (University of British Columbia, Canada) mainitsi idean roboteista pelaamassa jalkapalloa. Samaan aikaan Japanissa tutkijat perustivat työryhmän pureutumaan tekoälyn ongelmiin. Työryhmä leikki ajatuksella jossa jalkapallo—ottelu edistäisi tiedettä. Idea kehitty ja kesäkuussa 1993 ryhmä tutkijoita (Minoru Asada, Yasuo Kuniyoshi ja Hiroaki Kitano) ilmoittivat robottikilpailusta (Robot J-League). Kuukauden sisään ilmoituksesta idea oli saanut niin paljon kannatusta että päätettiin tehdä kilpailusta kansainvälinen. Projekti sai uuden nimen “Robot World Cup Initiative” tai lyhyesti ``RoboCup''. Syyskuussa 1993 tehtiin ensimmäinen virallinen ilmoitus aloitteesta ja 1997 pidettiin ensimmäinen virallinen RoboCup-ottelu.

The Robot World Cup Initiative (RoboCup) on kansainvälinen tutkimus— ja koulutusaloite jonka tarkoituksena on tuottaa älykkäitä robotteja, edistää teknologiaa ja kouluttaa. RoboCup:ssa on annettu ongelma, joka pitäisi ratkaista apuna kaikki mahdolliset teknologiat ja ongelmakentäksi on valittu jalkapallo. Tavoitteena on, että 2000–luvun puoliväliin mennessä itsenäisistä roboteista koostuva joukkue pelaa jalkapallo-ottelun, FIFA:n virallisilla säännöillä, viimeisintä maailmanliigan (World Cup) voittanutta joukkuetta vastaan.

Vuosittain järjetettävissä RoboCup: The Robot World Cup Soccer Games and Conferences —tilaisuudessa eri tahot kilpailuttava robottejaan ja ohjelmistojaan jalkapallossa. Jotta robottijoukkue voi toimia, täytyy ratkaista monenlaisia ongelmia: agenttien suunnitteluperiaatteita, monen agentin yhteistoiminta, startegian hankkiminen, reaaliaikajärkeily, robotiikka ja ympäristön aistimisen teknologia.

Robocupissa on kahdeksan liigaa:

- Simulator League, joka on ainoastaan erilaisille ohjelmille. Jokainen pelaaja on kuitenkin oma itsenäinen ohjelmansa (agentti). Kappaleessa 4.2 Käydään läpi Simulaattoriliigaa.
- Small Robot League, jossa kamppailee maksimissaan viisi alle parikymmentäsenttistä robottia joukkueessa pingispöydän kokoisella kentällä.
- Middle Size Robot League, josaa pelaa kaksi 7–11 robottista joukkuettä kentällä joka on 45–90 m leveä ja 90–120 m pitkä.
- Full Set Small Robot League, jossa on 11 joukkueessa robottia kilpailemassa.
- Sony Legged Robot League on Sonyn sponsoroima koirarobottitappelu.
- Humanoid League tulee olemaan vuodesta 2002 alkaen liiga, jossa robotit pelaavat jalkapalloa ihmisistä koostuvaa jalkapallojoukkuetta vastaan.

- TeleOperation Track, josta tulee informaatiota myöhemmin
- RoboCup Commentator Exhibition

4.2 Simulaattoriliiga

Alusta

Pelikentän tarjoaa Soccer Server palvelin, jonka tehtävänä on simuloida pelikenttää sekä pelaajien ja pallon liikettä. Asiakkaat kommunikoivat palvelimen kanssa UDP/IP sockettien avulla. Pelikenttä on kaksiulotteinen ja liikkuminen tapahtuu vain vaakatasossa. Peliä voidaan pelata joko pienellä kentällä tai standardikentällä. Pienellä kentällä on 1–5 pelaajaa joukkueessa ja standardikentällä on 6–11. Maalivahtieja ei ole koska pelaajia simuloidaan ympyroillä ja niillä ei ole käsiä.

Kaikki kommunikointi asiakkaiden kesken tapahtuu palvelimen kautta. Pelaaja lähettää sanoman käyttäen `say`—komentoa ja sanoma lähetetään broadcastina `hear` tietona. Sanoman pituus on maksimissaan 256 merkkiä ja palvelin sallii vain yhden sanoman asiakasta kohden yhden syklin aikana.

Ottelua on tuomitsemassa palvelimella tuomari. Se on vastuussa, jos on tullut maali, pallo ei ole enää kentällä ja se valvoo pelaajien sijaintia vapaapotku- sisäänheitto ja kulmapotkutilanteissa jottei puolustavan joukkueen pelaajat ole liian lähellä. Mukana on myös ihmistuomari tuomitsemassa epäurheilijamaista käytöstä /6/.

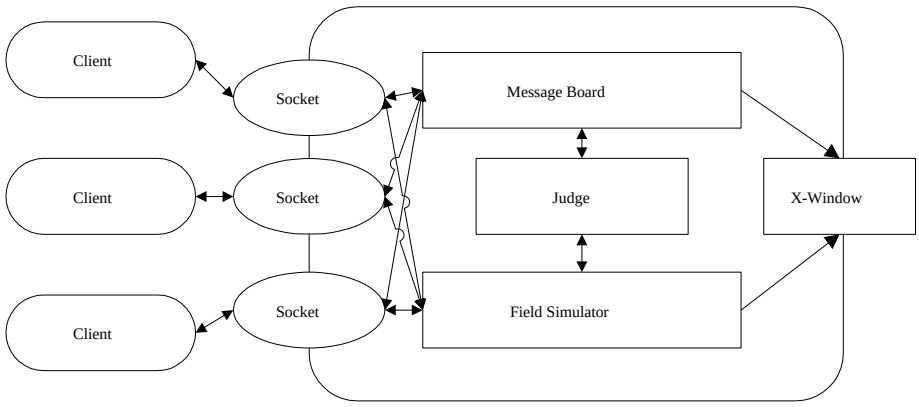

Figure 9. RoboCupin agenttialusta

5 Yhteenveto

On helppo ymmärtää agenttien merkitys hajautetuissa heterogeenisissä järjestelmissä. Niiden kyky tulla toimeen toisten moniagenttijärjestelmien agenttien kanssa ja kyky kommunikoida ontologioiden avulla auttavat luomaan yhtenäisyyttä heterogeenisiin hajautettuihin järjestelmiin. Agenttien mukautumis- ja oppimiskyky on omiaan muuttuvassa järjestelmässä.

Agentit eivät välttämättä ole uusi sovellusryhmä vaan ne voivat tuoda uutta tehoa jo olemassa oleviin sovelluksiin.Ne saattavat aluksi erottua omina sovelluksina, mutta ajan myötä ne muuttuvat osaksi käyttöjärjestelmiä tai sovelluksia.

Agenttiteknologia on kuitenkin yhä vielä vahvasti tutkimusala vaikkakin joitain agenttisovulluksia on jo olemassa. Olemassa olevat agentit eivät vielä toimi keskenään koska alalta puuttuu yhtenäinen konsensus siitä kuinka agentteja tulisi tukea käyttöjärjestelmätasolla. Vaikka aiheesta on runsaasti mieltänostattavaa mainospuhetta ei agentteja ole yleisesti hyväksytty väistämättömänä seuraavana askeleena hajautetussa teknologiassa.

/1/ Agent Working Group, *Agent Technology, Green paper*, http://www.omg.com 1999

/2/ Sasu Tarkoma, *Agenttien yhteistoiminta*, Helsinki 1999 Helsingin yliopisto

/3/ FIPA, *FIPA 97 Specification*, http://www.fipa.org 1998

/4/ FIPA, *FIPA 98 Specification*, http://www.fipa.org 1998

/5/ Tim Finin et. al., *Specification of the KQML Agent-Communication Language*, http://www.cs.umbc.edu/kqml/ 1993

/6/ Hiroaki Kitano et. al., *Robocup: The Robot World Cup Initiative*, http://www.robocup.org 1993