Pyry Lappalainen

Itsenäiset älykkäät agentit

Tietotekniikan kandidaatintutkielma

6. toukokuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Pyry Lappalainen

Yhteystiedot: pyjuella@student.jyu.fi

Ohjaaja: Timo Tiihonen

Työn nimi: Itsenäiset älykkäät agentit

Title in English: Autonomous intelligent agents

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 22+0

Tiivistelmä: Itsenäiset älykkäät agentit ovat fyysisiä tai virtuaalisia toimijoita, joita ohjaa tietokoneohjelma. Ne reagoivat ympäristön muutoksiin ja voivat itse vaikuttaa ympäristön tilaan. Kompleksiset agenttimallit ovat joustavampia, luotettavampia sekä monikäyttöisempiä, mutta samalla vaikeampia toteuttaa sekä vaativampia laskentatehon sekä muistinkäytön kannalta. Moniagenttijärjestelmillä voi hajauttaa monimutkaisen tehtävän pieniin osiin, joista huolehtivat itsessään yksinkertaisemmat agentit. Agentit ja agenttijärjestelmät ovat hyödyllisiä pitkäkestoisiin ja samanlaisina toistuviin tehtäviin. Hyvin yksinkertaiset tai lyhytkestoiset tehtävät saattavat olla helpompia tai halvempia toteuttaa vaihtoehtoisilla metodeilla.

Avainsanat: agentti, itsenäinen älykäs agentti, ohjelmistoagentti, moniagenttijärjestelmä

Abstract: Autonomous intelligent agents are either physical or virtual actors that are controlled by a computer program. They react to changes in their environment and can themselves affect the environment. Complex agent models are more flexible, reliable and multifunctional, but are also more difficult to implement and more demanding in terms of computing power and memory usage. Multiagent systems (MAS) can be used to split complex tasks into many simple ones, each of which are then handled by one or more internally simpler agents. Agents and agent systems are useful for long and repetetive tasks. Very simple or short tasks might be easier or cheaper to handle with alternative methods.

Keywords: agent, autonomous intelligent agent, software agent, multiagent system

Termiluettelo

Agentti	Ympäristössä oleva toimija, joka voi kerätä informaatiota ympäristöstä, sekä vaikuttaa siihen.
Ohjelmistoagentti	Agentti, jonka päätöksenteko perustuu tietokoneeseen ja ohjelmistoon.
Ympäristö	Ympäristö, jossa agentti toimii. Agentin näkökulmasta sen ympäristöön kuuluu kaikki mikä ei ole osa agenttia itseään, mutta voi olla vuorovaikutuksessa agentin kanssa.

Sisältö

1 Johdanto

Agentteja ja agenttijärjestelmiä voimme nähdä jokapäiväisessä elämässä. Esimerkiksi jääkaapin lämpötilaa ylläpitää yksinkertainen agentti. Itseohjautuvat autot ovat agentteja, joskin hyvin paljon monimutkaisempia. Internetselaimessa ilmestyviä mainoksia saattaa olla valitsemassa eräänlainen itsenäinen älykäs agentti, ohjelmistoagentti. Myös tietokonepeleissä voi ihmispelaajan vasta- tai kanssapelaajana toimia ohjelmistoagentti.

Itsenäiset älykkäät agentit ovat tekoälyteknologian osa-alue. Käytännössä tällaiset agentit ovat eräs tapa mallintaa itsenäisesti toimivia tekoäly-yksiköitä. Olio-ohjelmointiin perehtynyt henkilö voikin nähdä tiettyjä yhtenäisyyksiä agentin ja olio-ohjelmoinnin olion välillä. Itsenäiset älykkäät agentit toimivat saamansa syötteen perusteella, mutta kapseloivat, eli piilottavat, toimintatapansa vain itselleen.

Lukijalle saattaa herätä kysymys: Ovatko kaikki tekoälytoteutukset tai ohjelmat agentteja? Kysymykseen ei ole suoraa vastausta, sillä eri lähteet ovat antaneet agentille erilaiset määritelmät. Tässä tutkielmassa käytettävä määritelmä tarkennetaan 2. luvussa.

Tutkielmassa tutkitaan itsenäisiä älykkäitä agentteja. Tutkimusta lähdetään suorittamaan hyvin kevyeltä pohjalta ja siinä perehdytään eräisiin yleisimpiin agenttimalleihin. Tutkimusmenetelmänä on kirjallisuuskartoitus. Kartoituksessa tulee useassa kohtaa vastaan samoja tärkeita alan pohjateoksia, kuten Stuart Russelin ja Peter Norvigin *Artificial Intelligence: a Modern Approach* (käytössä ensimmäinen painos vuodelta 1995) ja Michael Wooldridgen *Multiagent Systems: an introduction* (2002). Suurin osa kartoituksessa löytyneistä lähteistä ovat suhteellisen vanhoja. Voidaankin päätellä, että tärkeimmät periaatteet ovat melko yleisesti hyväksyttyjä, ja että kehitys on suuntautunut näistä lähteistä periytyviin, yksityiskohtaisempiin malleihin, sekä agenttijärjestelmiin.

Tässä tutkielmassa ei esitetä konkreettisia toteutuksia edellä mainitun tapaisista agenteista, vaan tavoitteena on perehtyä eri mallien toimintaperiaatteisiin ja ymmärtää niitä abstraktilta pohjalta. Agentin toteuttamista tavoitteleva henkilö voi tämän tutkielman pohjalta kyetä valitsemaan itselleen tarkoituksenmukaisen mallin ja jatkaa tutkimusta kyseisen mallin osalta konkreettisen toteutuksen saavuttamiseksi.

Tutkielma koostuu viidestä luvusta. Toisessa luvussa määritellään tärkeät aihekohtaiset termit ja pyritään luomaan peruskäsitys agenttien yleisestä toimintamallista. Luvussa kolme käsitellään tarkemmin kuutta erilaista agenttimallia, ja neljännessä luvussa kahta erilaista agenttijärjestelmää, eli systeemiä, joka koostuu useasta agentista. Lopuksi yhteenvedossa pohditaan milloin on tarpeellista käyttää agenttia ja miten valita sopivan mallin.

2 Agentit ja ympäristö

Seuraavissa alaluvuissa määritellään käsitteet agentti, itsenäinen agentti, älykäs agentti sekä ympäristö. Lopuksi esitetään havainnollistavia esimerkkejä itsenäisistä älykkäistä agenteista.

2.1 Itsenäiset älykkäät agentit

Käsite itsenäinen älykäs agentti (eng. intelligent autonomous agent, lyh. ia) sisältää kolme alakäsitettä: itsenäinen, älykäs ja agentti. Nämä alakäsitteet kukin määrittelevät tärkeän ominaisuuden tutkielmassa käsiteltävästä agentista. Seuraavissa alaluvuissa määritellään nämä käsitteet.

2.1.1 Agentti

Agentti on toimija, joka kykenee havainnoimaan ympäristöään *sensoreidensa* avulla sekä vaikuttamaan ympäristönsä tilaan *toimilaitteillansa*. Ihmistoimijalla nämä sensorit vastaavat eri aisteja, ja vastaavasti kädet, jalat, suu ja muut elimet ihmisen toimilaitteita. Robottitoimijalla sensorit ovat kameroita, mikrofoneja ja infrapunatunnistimia. Toimilaitteina sillä on pyörät, mekaaniset raajat tai kaiuttimet. Ohjelmistoagentti saa havaintonsa sille toteutetusta syötemekanismista, ja se voi vaikuttaa virtuaaliseen ympäristöönsä esimerkiksi tulostusmekanismeilla tai olio-ohjelmoinnin tapoja noudattaen. (Russell ja Norvig 1995)

Tässä tutkielmassa keskitytään ohjelmiston pohjalta päätöksiä tekeviin agentteihin, jollaisia ovat esimerkiksi edellä mainitut robottitoimija sekä ohjelmistoagentti. Pelkästä agentista puhuttaessa voi lukija olettaa puhuttavan ohjelmistoon pohjautuvasta itsenäisestä älykkäästä agentista.

Wooldridge (2002, s.21) määrittelee kirjassaan agentin seuraavasti (kirjoittajan suomennos):

> ”Agentti on tietokonejärjestelmä, joka sijaitsee jossain ympäristössä, ja kykenee itsenäiseen toimintaan kyseisessä ympäristössä saavuttaakseen sille suunnitellun tavoitteen.”

Agentin itsenäisyys tarkoittaa, että sen päätöksentekoon ei vaikuta mikään ulkopuolinen tekijä, kuten esimerkiksi ihminen. Agentin toiminnan tehostamiseksi varhaisessa vaiheessa agenteille monesti syötetään pohjatietoa. Itsenäisen agentin on kuitenkin pohjatiedoistaan huolimatta perustettava tulevaisuuden päätökset sen omiin havaintoihin.

Itsenäisen agentin toimintaa ohjaa sille rakennettu *agenttiohjelma*. Agenttiohjelma toteuttaa agentin käyttötarkoitusta varten suunnitellun *agenttifunktion*, joka liittää syötteenä olevat havainnot ja kokemukset niitä vastaaviin toimintoihin. Agenttiohjelman yhdistää ympäristöön agentille toteutettu *arkkitehtuuri*. Arkkitehtuuri sisältää sensorit, jotka antavat agenttiohjelmalle tietoa ympäristön tilasta, sekä toimilaitteet, joilla agentti itse voi vaikuttaa ympäristön tilaan. Arkkitehtuuri saattaa myös sisältää ohjelmistoja, jotka toimivat agenttiohjelman ja sensoreiden välissä, mahdollistaen itse agenttiohjelman toteuttamisen korkeammalla tasolla. (Russell ja Norvig 1995)

2.1.2 Älykäs agentti

Wooldridge (2002) erottelee agentin sekä älykkään agentin. Älykkään agentin vaaditaan agentin määritelmän lisäksi olevan:

- Reagoiva – agentti reagoi ympäristössä tapahtuviin muutoksiin
- Ennakoiva – agentti pyrkii jatkuvasti saavuttamaan tavoitteensa
- Sosiaalinen – agentti on vuorovaikutuksessa muiden toimijoiden kanssa

Padgham ja Winikoff (2005) asettavat edeltävälle älykkään agentin määritelmälle vielä kaksi lisävaatimusta. Älykkään agentin on heidän mukaansa oltava lisäksi *joustava* – agentti voi saavuttaa tavoitteensa usealla eri tavalla – sekä *sinnikäs* – agentti kykenee jatkamaan toimintaansa epäonnistumisesta huolimatta.

Lisäksi Padgham ja Winikoff (2005) haluavat älykkään agentin tekävän päätöksiä järkevästi. Tässä yhteydessä on huomattava sanojen *älykäs* (eng. intelligent) ja *järkevä* (eng. rational) ero. Järkevä agentti toimii tavoitteitaan parhaiten edistävillä tavoilla, sekä pyrkii välttämään niille haitallisia toimia.

Wooldridge (2003) määrittelee järkeviä agentteja käsittelevässä kirjassaan agentin olevan jär-

kevä, mikäli agentti päättää suorittaa itselleen hyödyllisiä toimintoja, jotka perustuvat agentin sen hetkiseen uskomukseen ympäristöstä. Hän kuvailee seuraavan esimerkin (Wooldridge 2003, s.1, tutkielman kirjoittajan suomennos):

> ”Esimerkiksi, jos minulla on tavoite pysyä kuivana, ja uskon että ulkona sataa, on minun silloin järkevää ottaa sateenvarjo mukaan mennessäni ulos. Sateenvarjon ottaminen auttaa minua täyttämään kuivana pysymisen tavoitteen, ja on tässä mielessä parhaakseni. Vaikka erehtyisin ulkona satamisesta, olisi toimintaani silti pidettävä järkevänä; esimerkin pääasiana oli, että tein päätöksen, joka, jos uskomukseni maailmasta olisi ollut tosi, olisi auttanut minua saavuttamaan tavoitteeni.”

Oppivat agentit keräävät ympäristön niiden sensoreihin tuottamaa syötettä ja eri toiminnoista syntyneitä tuloksia muistiin. Tällaiset agentit voivat tämän ”historian” avulla tehostaa suorituskykyään ympäristössä. Oppivien agenttien käyttäytyminen voi siis muuttua niiden elinkaaren aikana.

2.2 Ympäristö

Ympäristö on agentin toiminta-avaruus. Useimmat agentit ovat suunniteltu toimimaan tietynlaisessa ympäristössä. Monimutkaisemmat agentit, kuten oppivat agentit voivat myös toimia niille tuntemattomissa ympäristöissä.

Ympäristöt voidaan luokitella niiden kompleksisuuden sekä ennustettavuuden perusteella. Agenttien toteuttamisen vaativuus ja niiden toimintavarmuus ympäristössä riippuu paljon tavoiteympäristöjen kompleksisuudesta. Russell ja Norvig (1995) jakavat ympäristön seuraavin perustein:

Tiedon saatavuus – Tieto voi olla ympäristössä joko avoimesti tai rajoitetusti saatavilla. Avoimessa ympäristössä agentti on välittömästi tietoinen ympäristön tilasta ja sen muutoksista. Usein, kuten oikeassa maailmassa toimivilla agenteilla, tiedon saanti ympäristöstä on rajoitettua.

Deterministisyys – Deterministinen ympäristö on ennustettavissa. Tällaisessa ympäristössä

jokainen toimi tuottaa ympäristöön aina tietyn, samanlaisen muutoksen. Epädeterministisessä ympäristössä voi esiintyä satunnaisuutta.

Muuttumattomuus – Muuttumattoman ympäristön tila ei muutu, ellei jokin toimija vaikuta siihen. Muuttuvan ympäristön tila taas voi vaihtua, vaikka yksikään toimija ei suorittaisi mitään toimintoja.

Jatkuvuus – Jatkumattomassa ympäristössä agentilla on rajoitettu määrä toimintoja, ja ympäristöllä rajoitettu määrä erilaisia tiloja.

2.3 Agentteja käytännössä

Useassa lähteessä käytetty esimerkki yksinkertaisesta fyysiseen maailmaan sijoittuvasta agentista on lämmityslaitteen termostaatti. Termostaatin tavoitteena on säilyttää huoneen lämpötila tietyssä arvossa. Termostaatissa on lämpötilasensori, jolla se saa tietoa ympäristöstään, eli huoneesta. Termostaatin toimilaite on itse lämmitin, jolla se voi vaikuttaa ympäristöön. Lämmittimen ollessa päällä huoneen lämpötila nousee, ja sen ollessa pois lämpötila laskee. Termostaatin logiikka on yksinkertainen: mikäli sensori kertoo lämpötilan olevan liian alhainen, on termostaatin saavuttaakseen tavoitteensa, nostettava lämpötilaa käynnistämällä lämmitin. Mikäli lämpötila on liian korkea, on lämmitin sammutettava.

Koska esimerkin agentti sijoittuu fyysiseen maailmaan, on sen ympäristö monimutkainen. Tieto on rajoitetusti saatavilla, sillä agentin havaintojen laatu ja määrä riippuvat esimerkiksi lämpötilasensorin sijoituspaikasta. Mikäli ulkolämpötila vaikuttaa huoneen lämpötilaan, on ympäristö säätilan takia muuttuva ja lisäksi käytännössä mahdoton ennustaa. Ympäristön jatkuvuus voi tämän agentin näkökulmasta olla kyseenalainen: agentille ympäristön tilaa kuvaa vain reaalilukuna esitettävä lämpötila. Teoriassa lämpötila voisi nousta äärettömän korkeaksi, mutta tilastolliselta kannalta se on mahdotonta.

Toisenlainen esimerkki agentista voisi olla tunnettua Pong-tietokonepeliä pelaava agentti. Pelissä pelaajina on kaksi suorakulmaista ”mailaa”, joiden tavoitteena on ylös ja alas liikkumalla estää niiden välissä kimpoilevan pallon osumista omiin maaleihinsa. Mailaa ohjaava agentti saa havaintoinaan pallon sijainnin ja nopeuden. Agentti sitten liikuttaa mailaa asete-

tulla nopeudella sitä paikkaa kohti, jossa se on pallon laskenut osuvan maaliinsa.

Tässä tapauksessa agentin ympäristö sijoittuukin virtuaaliseen maailmaan, ja on siten agentin kannalta helpompi. Tieto on välittömästi agentin saatavilla: se saa syötteenään reaaliajassa pallon sijainnin ja nopeuden. Ympäristö on ennustettavissa, sillä agentti voi tarkasti laskea, mihin suuntaan pallo kimpoaa ja millä nopeudella, sekä myös, ehtiikö se torjua omaa maaliaan kohti liikkuvan pallon. Vaikka ympäristöllä on hyvin paljon erilaisia mahdollisia tiloja, on niitä silti rajattu määrä. Tämäkin ympäristö on kuitenkin muuttuva, sillä pallo liikkuu mailoista huolimatta.

3 Yleisiä agenttimalleja

Erilaisia agenttimallien toteutuksia on monia, joten niitä on tarkoituksenmukaista käsitellä jonkin jäsentyneen jaottelun avulla. Eräs jaotteluperuste eri agenttien malleille on niiden älykkyyden ja itsenäisyyden taso. Russell ja Norvig (1995) jakavat agentit seuraaviin luokkiin:

- Refleksiagentit
- Mallipohjaiset refleksiagentit
- Tavoitepohjaiset agentit
- Hyötypohjaiset agentit

Seuraavissa alaluvuissa käsitellään aluksi näitä neljää Russelin ja Norvigin määrittelemää luokkaa, ja lisäksi kahta muuta yleistä agenttimallia: BDI-agentit ja oppivat agentit. Koska oppivat agentit on hyvin laaja käsite, pysyttäydytään tässä tekstissä Russelin ja Norvigin esittelemässä oppivien agenttien mallissa.

Osaa malleista havainnollistetaan pelkistetyillä esimerkeillä, jotka, ellei tekijää erikseen mainita, ovat tutkielman kirjoittajan kehittämiä. Lukijan on otettava huomioon, että esimerkit eivät välttämättä vastaa olemassa olevia agentteja, vaan niiden tarkoituksena on selventää kyseisen agenttimallin toimintaperiaatetta.

3.1 Refleksiagentit

Refleksiagentit (eng. reflex agent) ovat malleista yksinkertaisimpia. Refleksiagentit eivät pidä kirjaa tapahtumista, vaan toimivat täsmälleen kyseisen hetken havaintojen mukaisesti. Refleksiagenteilla on sisäänrakennettu joukko sääntöjä, joista ne valitsevat parhaan havaintojensa perusteella. Valittu sääntö yhdistyy sitten toimintoon, joka agenttien tulee suorittaa. Agenttien säännöt on helppo kuvitella ehtolauseina: ”jos ehto E_x toteutuu, suorita toiminto T_x”. (Russell ja Norvig 1995)

Yksinkertaisissa tehtävissä, kuten lämmityslaite -esimerkissä, refleksiagentit ovat toimivia, laskennan kannalta halpoja ja suhteellisen helppoja toteuttaa. Monimutkaisemmissa tehtävis-

sä yksinkertaiset refleksiagentit saattavat rajoitteidensa takia menettää suorituskykyään osittain tai täysin. Vastaavasti monimutkaisen refleksiagentin toteutuksessa sääntökokoelman, ja siten toimintaa ohjaavan agenttiohjelman, koko kasvaa huomattavasti, mikä saattaa johtaa lisäongelmiin.

Tutkitaan seuraavaa esimerkkiä refleksiagentista: sovitaan että agentin tehtävä on hakea vettä kaivosta ja kaataa se saaviin. Agentilla on seuraavat säännöt:

- Jos agentti on saavilla, on sen kaadettava vesi saaviin ja sitten mentävä kaivolle
- Jos agentti on kaivolla, on sen täytettävä ämpäri, ja ämpärin täytyttyä mentävä takaisin saaville

Ideaalitapauksessa agentti kulkee väliä edestakaisin ja saavi täyttyy. Valitettavasti kuuma keli on kuivattanut kaivon, eikä ämpäri täyty kaivolla käydessä. Koska refleksiagentti ei tiedä kaivon tilaa tai miten se muuttuu, jatkaa se ämpärin täyttämistä loputtomasti. Tässä tilanteessa refleksiagentti on jumissa, eikä saavi täyty, vaikka agentti jatkaa ahkeroimistaan.

Esimerkin agenttia voisi kehittää antamalla sille uuden säännön: mikäli agentti on kaivolla, ja kaivo on kuiva, on agentin mentävä pidempää reittiä järvelle ja täytettävä ämpäri siellä. Päivityksestä seuraa kuitenkin ongelma: refleksiagentin on käytävä kaivolla joka kerta, vaikka se olisi edelliselläkin kerralla ollut tyhjä.

3.2 Mallipohjaiset refleksiagentit

Mallipohjaiset refleksiagentit (eng. model-based agent) sisältävät sääntökokoelman lisäksi ylläpidettävän mallin ympäristön tilasta. Sisäisesti säilötyn tilan päivittämiseksi on agentin tiedettävä, kuinka tila kehittyy agentista itsenäisesti, sekä kuinka agentin omat toiminnot vaikuttavat tilaan. Näiden tietojen antaminen agentille on agentin ohjelmoijan vastuulla.

Valitessaan toimintoa mallipohjainen refleksiagentti toimii lähes samoin kuin refleksiagentti, mutta ennen säännön valitsemista se päivittää sisäisen tilansa. Tilan päivittämisessä agentti ottaa huomioon aiemman sisäisen tilan, tiedon sen kehittymisestä, sekä hetkelliset havainnot. Päivittämisen jälkeen agentti valitsee säännön uuden sisäisen tilansa perusteella. (Russell ja Norvig 1995)

Jatketaan päivitettyä kaivoesimerkkiä: sääntökokoelma on sama kuin aiemmin päivitetyllä refleksiagentilla, mutta agentilla on lisäksi sisäinen malli ympäristön tilasta, sekä käsitys sen kehittymisestä. Tällöin agentti voi saavilta lähtiessään muistaa ”on kuuma, ja kaivo oli viime kerralla kuiva” ja päätellä ”kaivo on siis nytkin kuiva, menen suoraan järvelle”. Lisäksi, jos alkaa sataa, voi agentti päivittäessään sisäistä tilaansa päätellä, että sateen johteesta kaivo ei ole enää kuiva, ja näin parantaa suorituskykyään kulkemalla taas lähempänä olevalle kaivolle.

3.3 Tavoitepohjaiset agentit

Aina ympäristön tilan ja sen seurausten tietäminen ei ole riittävää. Otetaan esimerkiksi mallipohjaiseen agenttiin perustuva robottiauto, joka osaa ajaa tiellä, kääntyä ja jarruttaa. Agentin ajaessa risteykseen, on sen kuitenkin tehtävä päätös: mihin suuntaan mennä? Mallipohjainen agentti osaisi välttää suoraan jatkuvalla tiellä olevan tietyön, mutta se ei tietäisi kääntyäkö vasemmalle tai oikealle. Ongelmaa voisi ehkäistä määräämällä agentin kääntymään aina mieluiten oikealle, tai käyttämällä satunnaisuutta. Kumpikin toteutus tekisi kuitenkin agentista melko hyödyttömän: joko agentti ajaisi ympyrää, tai se päätyisi minne sattuu. Tämän vuoksi agentilla tulisi olla määränpää, tai yleisemmin, tavoite.

Suurin ero tavoitepohjaisella agentilla (eng. goal-based agent) edellisiin malleihin on valmiiksi tehtyjen sääntöjen puuttuminen. Tavoitepohjainen agentti tekee itse sääntönsä tilannekohtaisesti, sen tavoitetta parhaiten edistävällä tavalla. Päivittäessään sisäistä tilaansa tavoitepohjainen agentti pyrkii päättelemään, millaiseen tilaan sen kukin siinä tilanteessa mahdollinen toiminto saattaisi ympäristön ja agentin itsensä, ja mikä niistä edistäisi agentin tavoitetta eniten. (Russell ja Norvig 1995)

Laskennallisesti tavoitepohjainen agentti on tietokoneelle edellisiä raskaampi, mutta useassa tapauksessa tavoitepohjainen agentti on paljon monikäyttöisempi ja joustavampi. Refleksiagentti voi tehdä tietyn tehtävän halvemmalla monta kertaa, mutta tehtävän muuttuessa on sen säännöstö rakennettava uudelleen uuden tehtävän mukaiseksi. Ongelmatapauksia varten on refleksiagentille taas koodattava useita tilannekohtaisia sääntöjä. Tavoitepohjaisella agentilla toisen samanlaisen tehtävän onnistumiseksi vaaditaan usein vain tavoitteen muuttaminen.

Näin tavoitepohjaisen agentin ohjelmiston pituus ja muistinkäyttö voivatkin olla pienempiä.

3.4 Hyötypohjaiset agentit

Aina pelkkä tavoite ei takaa agentille parasta mahdollista käyttäytymistä. Jotkut tavat saavuttaa tavoitteita voivat olla hitaampia, riskialttiimpia, kalliimpia tai muuten ”epämiellyttävämpiä” kuin toiset. Esimerkiksi agentin valitsema reitti tavoitteeseen voi olla lyhin, mutta siinä on suurempi riski epäonnistua kuin pidemmässä reitissä. Vastaavan kaltaisissa tapauksissa on tärkeää voida vertailla kuinka ”tyytyväiseksi” eri tilat tai tapahtumat voivat agentin saattaa. (Russell ja Norvig 1995)

Tämä ”tyytyväisyys” kertoo, kuinka paljon *hyötyä* (eng. utility) tilasta tai tapahtumasta agentille on. Konkreettisesti tilan tai tapahtuman hyöty voi olla vain jokin reaaliluku. Tällöin agentti pyrkii suorittamaan toiminnot, joista yhteensä syntyvä hyötyarvo on kaikista suurin. Hyötyarvot auttavat agenttia myös tekemään järkeviä päätöksiä, jos agentilla on useita eri tavoitteita, joiden onnistuneesti saavuttaminen ei ole täysin varmaa. Vastaavasti, mikäli agentti on tilanteessa, jossa se vääjäämättä joutuu suorittamaan negatiivisen hyötyarvon omaavan toiminnon, kykenee se valitsemaan toiminnoista vähiten haitallisen.

Hyötypohjainen agentti (eng. utility-based agent) vaatii toimiakseen sille suunniteltuja eri tiloihin liitettyjä hyötyarvoja. Nämä arvot on ohjelmoitava agenttiohjelmaan suunnittelu- tai ohjelmointivaiheessa. Hyötyarvojen tallentamisesta johtuen hyötypohjaisen agentin muistinkäyttö on tavoitepohjaista suurempi, ja agentin ohjelmoiminen on vastaavasti yhtä muuttujaa monimutkaisempaa. Hyötypohjainen agentti voi kuitenkin haastavammassa ympäristössä kyetä toimimaan varmemmin ja tehokkaammin kuin edellä käsitellyt mallit.

3.5 Oppivat agentit

Oppiva agentti ei itsessään ole uusi agenttiluokka, vaan se sisältää jonkin edellä käsitellyistä malleista toimivana osanaan. Toimivan osan lisäksi oppiva agentti sisältää *arvioivan osan*, *oppivan osan* sekä *ongelmageneraattorin*. (Russell ja Norvig 1995)

Oppiva osa on agentin elementti, joka varsinaisesti muuttaa toimivan osan sääntöjä ja tapoja.

Oppiva osa käyttää omaa tietoaan ja arvioivan osan antamaa palautetta päättäessään mitä muutoksia toimivaan osaan tulisi tehdä, jotta suorituskyky paranisi.

Arvioivan osan tehtävä on välittää oppivalle osalle, kuinka hyvin agentti pärjää. Arvioiva osa saa tietoa agentin sensoreista ja se voi verrata tätä tietoa agentin kiinnitettyyn toimintastandardiin (vastaa hyötypohjaisen agentin hyötyarvoja). On tärkeää, että oppiva agentti ei itse voi muuttaa standardejaan, sillä muussa tapauksessa agentti voisi vain päättää olevansa tyytyväinen nykyiseen tilaansa sen edistämisen sijaan.

Ongelmageneraattori ehdottaa toimivan osan valittuihin toimintoihin pieniä muutoksia. Nämä muutokset auttavat oppimisprosessia tilanteissa, joissa agentti luulee jo toimivansa hyvin, mutta voisi toimia tehokkaamminkin. Ongelmageneraattorin tuottamista muutoksista voi arvioiva osa muutetun toiminnon jälkeen päätellä, oliko muutoksesta hyötyä vai haittaa.

3.6 BDI-agentit

Joskus ympäristön tieto ei ole täydellisesti saatavilla ja tiedon sekä toimintatavan päivittäminen jatkuvasti on liian kallista. Toisaalta tietyn, aiemmin tehdyn toimintasuunnitelman tiukka noudattaminen saattaakin nykyisessä tilanteessa johtaa epäonnistumiseen.

Tällaisissa tapauksissa, olettaen, että tavoitteisiin vaikuttavat merkittävät muutokset voidaan määrittää välittömästi, uskomuksiin perustuvien agenttien käyttö voi vähentää toimintasuunnitelman päivittämistarvetta huomattavasti (Rao ja Georgeff 1995).

Bratman (1987) esittelevät kirjassaan BDI-arkkitehtuurin, joka on eräs tapa mallintaa älykkäitä itsenäisiä agentteja. Lyhenne BDI tulee englanninkielisistä sanoista *belief* (uskomus), *desire* (halu) ja *intention* (aikomus).

Uskomus on BDI-agentilla oleva kuva ympäristön tämänhetkisestä tilasta. Sananvalinnalla painotetaan, että agentin maailmankuva ei välttämättä ole tosi. Tämä tarkoittaa, että agentilla oleva tieto voi esimerkiksi olla vanhentunut, rajoittunut tai agentti voi olla tehnyt virheellisen havainnon.

Halu on BDI-agentin pitkän tähtäimen tavoite. Halusta erotetaan lyhyen tähtäimen tavoite

(eng. goal), joka on agentin aktiivisesti toteutettavaksi valitsema halu. Agentin valitsemat lyhyen tähtäimen tavoitteet, toisin kuin halut, eivät voi olla ristiriidassa. Mikäli agentilla olisi haluina ”pysy lähellä paikkaa A” ja ”hae esine paikasta B”, voisi vain toinen näistä olla aktiivisena.

Aikomukset ovat BDI-agentin korkean tason suunnitelmia, jota se noudattamaa saavuttaakseen maalinsa. Agentin on tarkennettava aikomuksensa yhä pienemmiksi osiksi, kunnes ne ovat yksittäisiä, toteutettavia toimintoja. Agentti on sitoutunut valitsemiinsa aikomuksiin, ja se seuraa niitä, kunnes maalit on saavutettu. Bratmanin mukaan agentti voi hylätä aikomuksensa ainoastaan, jos se toteaa aikomuksen täysin mahdottomaksi toteuttaa, tai aikomus on edeltävä toiselle, hylättävälle aikomukselle. Cohen ja Levesque (1990) määrittelevät Bratmanin mukaisen agentin *fanaattisesti* aikomuksiinsa sitoutuneeksi. He määrittelevät myös *suhteellisesti* sitoutuneen agentin, joka tietyissä olosuhteissa voi hylätä aikomuksensa.

Koska aikomukset voivat käsittää pitkän aikavälin ennen maalin toteutumista, ei agentti voi tarkentaa niitä välittömästi. Pitkälle katsottaessa eri valintoja voi olla yksinkertaisesti liikaa agentille rajatussa muistissa ja niiden laskeminen voi kuluttaa huomattavasti aikaa ja laskentatehoa. Lisäksi, kuten aiemmin mainittiin, saattaa aikaisempiin ja mahdollisesti vanhentuneisiin, uskomuksiin perustuva suunnitelma johtaa epäonnistumiseen nykytilassa. Tämä johtaakin kehittäjän vastattavaan kysymykseen: milloin ja kuinka pitkälle agentin kannattaa tarkentaa aikomuksiaan?

4 Moniagenttijärjestelmät ja kerrosarkkitehtuurit

Edellisessä luvussa läpi käydyt mallit ovat kaikki selvästi yksittäisiä toimijoita. Näitä toimijoita voidaan tarvittaessa yhdistää useammasta eri mallin agentista koostuvaksi järjestelmäksi, jossa toimijat ovat vuorovaikutuksessa keskenään, ja toimivat yhteistä tavoitetta kohti. Seuraavissa alaluvuissa käsitellään kahta tällaista mallia: moniagenttijärjestelmää, jossa järjestelmän sisältämät agentit ovat selvästi itsenäisiä, eroteltavia toimijoita, sekä kerrosarkkitehtuuria, jossa agentin päätöksentekoon vaikuttaa monta eri tasolla toimivaa sisäistä toimijaa.

4.1 Moniagenttijärjestelmät

Moniagenttijärjestelmät (eng. multiagent systems, MAS) ovat nimensä mukaan useista agenteista koostuvia järjestelmiä. Tällaiset järjestelmät sisältävät useita samoja tai eri tehtäviä tekeviä agentteja, jotka kommunikoivat ja tekevät yhteistyötä toistensa kanssa. Yhteistyön avulla agentit voivat kyetä toimiin, joihin yksittäinen agentti ei kykenisi tai se olisi huomattavasti monimutkaisempi toteuttaa. (Wooldridge 2002)

4.1.1 Määritelmä

Moniagenttijärjestelmissä tieto koko ongelmasta ja sen ratkaisukyky on jaettu järjestelmän agenttien kesken. Agentit suorittavat omaa tehtäväänsä itsenäisesti ja kommunikoivat tarvittaessa toisilleen. Sycara (1998) määrittelee moniagenttijärjestelmän seuraavin perustein:

- Jokaisen yksittäisen toimijan tieto sekä kyvyt ongelman ratkaisemiseksi ovat epätäydellisiä.
- Järjestelmässä ei ole globaalia ohjausjärjestelmää
- Tieto on hajautettua
- Laskenta on epäsynkronoitua

4.1.2 Hyötyjä ja heikkouksia

Moniagenttijärjestelmien käyttö voi vähentää laskentatehon tarvetta, sillä kaikkien agenttien ei tarvitse olla aktiivisena koko aikaa. Järjestelmään ilmeneviä pullonkaulailmiötä voi lieventää lisäämällä vaativaa tehtävää tekevien agenttien määrää. Lisäksi järjestelmän osia on sen modulaarisuuden vuoksi helpompi päivittää tai muokata, ja tarvittaessa osia järjestelmästä uudelleenkäyttää toisessa järjestelmässä. (Rudowsky 2004)

Moniagenttijärjestelmä voi monimutkaisessa ja pitkäaikaisessa tehtävässä olla varsin toimiva. Tällaisen järjestelmän toteuttaminen vaatii kuitenkin tehtävään ja ympäristöön pohjautuvaa perinpohjaista suunnittelua, jotta järjestelmän kaikki osat toimisivat saumattomasti yhdessä. Tästä syystä myös yksinkertaista tehtävää varten on harvoin järkevää toteuttaa moniagenttijärjestelmää.

4.2 Kerrosarkkitehtuurit

Usein agentilta vaaditaan sekä refleksimäistä (nopeaa, hetkellistä) että ennakoivaa (pitkän ajan käsittävää) käyttäytymistä, jollaista ei välttämättä saavuteta aiemmin esitellyillä yksittäisten agenttien malleilla. Tällöin eräs ratkaisu voisi olla käyttää moniagenttijärjestelmää, mutta yksinkertaisemmissa tapauksissa sellainen saattaisi olla tarpeettoman monimutkainen.

Yksi vaihtoehto moniagenttijärjestelmälle on kerrosarkkitehtuuri. Kerrosarkkitehtuurissa on useampia kerroksiin jaettuja alajärjestelmiä, jotka huolehtivat omalle tasolleen sopivasta käytöksestä. Usein kerroksia on kaksi, yksi refleksimäiselle ja yksi ennakoivalle, mutta kerroksia voi toteutuksesta ja tarkoituksesta riippuen olla useampiakin. Weiss (2013) jakaa kerrosarkkitehtuurin kahteen osaan kerroksien hallintahierarkian mukaan: *vierekkäin* kerrostetut ja *päällekkäin* kerrostetut.

Vierekkäin kerrostetussa arkkitehtuurissa kaikki kerrokset on yhdistetty agentin sensoreihin ja toimilaitteisiin, ja tuottavat ehdotuksia näkökulmastaan oikealle toiminnolle. Koska ehdotukset voivat olla hyvin eroavaisia, saattaa agentin käyttäytyminen olla vaihtelevaa. Tästä johtuen kerrostettu arkkitehtuuri tarvitsee myös keskuspäätöksentekijän, joka päättää johdonmukaisesti mitä näistä ehdotuksista noudatetaan. Tämä päättäjä aiheuttaa myös

keskeisimmät ongelmat vierekkäisessä kerrostuksessa: päättäjälle tulevien ehdotusten kombinaatioiden määrä kasvaa eksponentiaalisesti kerroksia ja kerroksien eri ehdotusvaihtoehtojen määrää kohtaan. Lisäksi keskitetty päätöksentekijä saattaa aiheuttaa pullonkaulailmiön agentin päätöksentekoprosessissa. (Weiss 2013)

Päällekkäin kerrostetuissa arkkitehtuureissa agentin sensoreihin ja toimilaitteisiin yhdistyy kuhunkin korkeintaan yksi kerros. Agentin tietovirta kulkee siten jokaisen kerroksen kautta. Tietovirta voi kulkea joko *yksivaiheisesti* tai *kaksivaiheisesti*. Yksivaiheisessa virrassa tieto kulkee sensoreista lähtien jokaisen kerroksen läpi, ja viimeinen kerros suorittaa itse toiminnan. Kaksivaiheisessa virrassa tieto kulkee vastaavasti sensoreista lähtien hierarkian alapäästä sen ylätasolle asti, josta palautuu vastaavat ohjauskomennot takaisin hierarkian alapäähän. (Weiss 2013)

Kerrosarkkitehtuuri ei ole moniagenttijärjestelmä, eikä se täytä moniagenttijärjestelmän määritelmää. Kerrostettu agentti toimii selvemmin yksikkönä, eikä sen tietojen ja taitojen tarvitse olla hajautettuja.

5 Yhteenveto

Tutkielmassa perehdyttiin useaan erilaiseen malliin itsenäisestä älykkäästä agentista. Tutkielmassa esiteltyjen mallien lisäksi on olemassa vielä paljon erillisiä agenttimalleja sekä jatkokehitelmiä tutkielman malleista. Tämän tutkielman pohjalta on saatu kuitenkin perustason ymmärrys yleisistä agenteista, ja siten kyky ymmärtää uusienkin agenttimallien toimintaperiaatetta ja toteutusta.

Tutkielmasta saaduilla tiedoilla voi myös oman agentin kehittämistä suunnitteleva lukija saada suunnittelulle ja kehitykselle suuntaa. Tärkeimpänä kysymyksenä agentin kehittämisessä on agentin tehtävä: millaiseen ympäristöön, ja mihin tarkoitukseen agentti luodaan? Tähän yleispätevästi hyvä vastaus lienee valita yksinkertaisin tehtävässä toimiva malli. Yksinkertaisen mallin toteuttaminen on varsinkin uudelle kehittäjälle helpompaa, ja lisäksi koko suunnittelu- ja toteuttamisprosessit ovat helpompia. Yksinkertainen agentti on myös toiminnan aikana laskennallisesti vähemmän vaativa. Mikäli agentista halutaan tehdä monikäyttöisempi, on valittava tarvittavaa kompleksisempi malli, vaikka alkuperäinen ympäristö ja tehtävä olisivatkin yksinkertaisia. Moniagenttijärjestelmän tai kerrosarkkitehtuuriagentin kaltaisia toteutuksia kannattanee harkita vasta todella monimutkaisen, monitasoisen tai suurikokoisen tehtävän edessä.

Ennen agentin suunnittelua kannattaa kehittäjän kuitenkin miettiä, vaatiiko tehtävä juuri agenttia, vai riittäisikö jokin vaihtoehtoinen toteutus, kuten tehtävän käsin suorittaminen tai esimerkiksi ohjelmistotehtävässä perinteinen ohjelma. Agenttimainen toteutus on itsenäinen, automatisoituva ja pitkäkestoinen, mutta sen toteuttaminen voi, varsinkin lyhytaikaista tehtävää varten, olla vaihtoehtoista toteutusta työläämpää. Tämä korostuu etenkin, kun kehittäjällä on vaihtoehtoisesta tavasta enemmän kokemusta kuin agenttimaisesta.

Voidaan siis tehdä johtopäätökset: itsenäinen älykäs agentti on hyvä pitkäkestoisessa, toistuvassa, mutta tapahtumamahdollisuuksiltaan rajatuissa tehtävissä. Tällaiseen tehtävään sopivasti valittu agenttimalli on toteutettuna tehokas ja pitkäikäinen. Moniagenttijärjestelmät ovat päteviä suuriskaalaisissa, mutta selkeästi jaoteltavissa tehtävissä. Hyvin yksinkertaiset ja lyhytkestoiset tehtävät voi olla parempi suorittaa vaihtoehtoisella metodilla.

Lähteet

Bratman, Michael. 1987. *Intention, plans, and practical reason.* Nide 10. Harvard University Press Cambridge, MA.

Cohen, Philip R, ja Hector J Levesque. 1990. “Intention is choice with commitment”. *Artificial intelligence* 42.

Padgham, Lin, ja Michael Winikoff. 2005. *Developing intelligent agent systems: A practical guide.* Nide 13. John Wiley & Sons.

Rao, Anand S, ja Michael P Georgeff. 1995. “BDI agents: from theory to practice.” Teoksessa *ICMAS,* 95:312–319.

Rudowsky, Ira. 2004. “Intelligent agents”. *Communications of the Association for Information Systems* 14:14.

Russell, Stuart J, ja Peter Norvig. 1995. *Artificial intelligence: a modern approach.* Prentice Hall International.

Sycara, Katia P. 1998. “Multiagent systems”. *AI magazine* 19:79.

Weiss, Gerhard. 2013. *Multiagent systems.* MIT press.

Wooldridge, Michael. 2002. *Multi-agent systems: an introduction.* Wiley.

———. 2003. *Reasoning about rational agents.* MIT press.